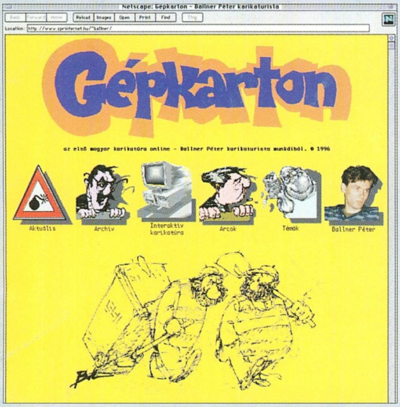


A hónap Internet ajánlata: Gépkarton

ára: 190 Ft

infopen

nyílt rendszerek magazinja
V. évf. 3. szám 1997. március

- Hogyan készítsünk saját honlapot a Weben?
- Az objektumokról informatikai vezetőknek
- Informix Illustra
- Hálózatkezelés Jávában
- IBM hálózati szolgáltató központ
- Interjú az IBS alelnökével
- Sun Unix mainframe
- WWW- és könyvajánlatok

„A hálózati számítástechnika letéteményese”



Stewart R. Oldroyd, az Oracle Hungary ügyvezető igazgatója

A hálózati számítástechnika letéteményese

Oracle, a stratégiai partner

Korszerű adatbázis-kezelők, fejlesztőeszközök, komplex alkalmazások, webes verziók megjelentetése, a Network Computer koncepciójának kidolgozása, bevezetésének hathatós támogatása — mind egy-egy fontos állomás a töretlen fejlődést megvalósító, felhasználói szempontból egyre inkább a stratégiai partner szerepét játszó Oracle Corporation működésében. Mindezekről és a magyarországi leányvállalat tevékenységéről *Stewart R. Oldroyd* ügyvezető igazgatót kérdeztük.



Az Oracle az utóbbi időben kezdeményező, újító szerepet vállalt az informatika világában. Mind technológiájában, mind piaci működésében folyamatos átalakuláson megy keresztül. Melyek a főbb stratégiai elemei ennek a változásnak?

S. O.: Három fő stratégiai elemet emelnék ki. Az első az, hogy jól ismert adatbázis-szállítóból egyre inkább a felhasználók stratégiai partnerévé szeretnénk válni. Ehhez megfelelő alapot teremt cégünk fejlődése és termékskálánk jelentős kiszélesedése. Egyre fejlettebb, egyre többet tudó adatbázis-kezelőink mellett megjelentek és nagyon sikeresek Designer/2000, Developer-2000 és Discoverer fejlesztőeszközeink, Oracle Financials, Oracle Manufacturing alkalmazásaink, Oracle Express vezetői információs, döntéselőkészítő rendszereink vagy már a hálózati számítógépeken is futtatható Oracle InterOffice irodaautomatizációs alkalmazásunk. Mindezekkel együtt előrehaladtunk az olyan új technológiai területeken, mint a Data Warehousing, az objektumorientáltság, az Internet és az intranet lehetőségeinek a kihasználása.

A felhasználóknak az egész céget átfogó, az igényeket komplex módon kielégítő, a legkorszerűbb technológiákat felvonultató, integrált vállalati rendszerre van szükségük. Ez mindenképpen új piacokat és új feladatokat jelent számunkra. Így például, ellentétben a korábbi évekkel, amikor licenceladásainkhoz elegendő volt a felhasználó néhány szakemberével, az informatikai vezetővel, adatbázis-adminisztrátorral és alkalmazásfejlesztővel tárgyalnunk, egyeztetnünk, igényeiket felmérnünk, ezután mint stratégiai partnernek meg kell ismernünk a felhasználó ügyfeleinek az elvárásait, sőt konkurenseinek működését, szereplését is.

E munka magas színvonalú elvégzéséhez az Oracle Corporation szervezetén belül, az egész világot átfogóan hozunk létre a vertikális piaci területekkel, piaci szektorokkal foglalkozó, azokat kutató, ismerő szakértői csoportokat. Az első ilyen szakértői csoportok az olaj- és energiaipar, a banki-biztosítási szektor, a távközlés és az államigazgatás terén kezdik meg működésüket.

Az Oracle átalakulásában a következő fontos stratégiai elem a cég ismertségének nagyarányú kiterjesztése. Korábban természetes volt, hogy mint adatbázisrendszer-szállító az Oracle csak a felhasználók informatikai szakemberei előtt volt ismert. Különösen így volt ez az alkalmazásszállító által installált rendszerek esetében, amikor az Oracle fejlesztőeszközeivel és adatbázis-kezelőjére alapozva készültek el ugyan az alkalmazások, de a felhasználóknál ez alig került nyilvánosságra. Mint új stratégiai partner, cégünk az eddigieknél sokkal szélesebb körben és lényegesen kiemeltebb, meghatározóbb szerepet ellátva fejti ki piaci tevékenységét. Ez részben növeli a márkanév ismertségét, ám egyúttal igényli is, hogy megfelelő marketingeszközökkel elősegítsük azt. Érdekességként említem, hogy 1992-ben a Wall Street Journal olvasói között, tehát nem elsősorban informatikai szakemberek körében végzett felmérés szerint az Oracle cég ismertsége néhány százalékos volt. Egy 1996 közepén ugyanezen körben megismételt felmérés már 70 százalékos ismertséget mutatott, annak a kezdeményező szerepnek köszönhetően, amit cégünk az Internet/intranet kihasználása, a Network Computer koncepciójának kidolgozása terén játszott.

Épp ide kapcsolódik a harmadik stratégiai elem, a hálózati számítástechnika, mint az informatika világának olyan új és egyre inkább meghatározó létformája, aminek az általánossá válásában az Oracle az NC-technológia kidolgozójaként döntő szerepet szán magának.

Mi a hálózati számítástechnika lényege, miért jó, ha az NC-k felváltják a PC-ket?

S. O.: A hálózati számítástechnika általánossá válása megsokszorozza az informatikában rejlő lehetőségeket, csökkenti a számítástechnikai költségeket, esélyegyenlőséget teremt országok, vállalatok, kis és nagy cégek között stb.

A PC-vel kapcsolatban: bármilyen jó eszköz is a személyi számítógép, bármilyen nagy előrelépés volt a nagygépes technikát követően a megjelenése, mégis úgy hiszem, a több mint 15 éves PC-korszak a végéhez közeledik. A PC-knek ugyanis legalább három lényeges hátrányuk van az NC-khez képest. Az egyik, hogy drágák, magas a beszerzési áruk, és ha már megvannak, akkor is folyamatos bővítésre szorulnak. Másodszor, folyamatos karbantartást igényelnek az alapszoftvereket és az alkalmazásokat illetően. Jelentős éves kiadás az új szoftververziók beszerzése, installálása. S végül, egy ugyancsak lényeges szempont — ami az egyik akadálya az informatika nagyobb intenzitású, világméretű terjedésének — a PC-k, a Windows operációs rendszerek bonyolultsága.

Ezzel szemben az NC olcsó, elmaradnak a bővítésekre és az új szoftververziókra vonatkozó költségek, és alkalmazása is rendkívül egyszerű. Nem beszélve arról, hogy egy azonosító kártya használatával bármely személy, bárhol a világban, bárkinek az NC-jén keresztül azonosítva magát, már barangolhat is a hálózaton. Belátható időn belül megvalósulhat,

hogy az utcai telefonfülkékhez hasonlóan, az NC-fülkékből is kapcsolatot lehet majd teremteni az Internettel.

Természetesen az NC-technológia általánossá válásához szükség van bizonyos szemléletváltásra is. Mindenkinek meg kell szoknia azt a gondolatot, hogy személyes levelei, adatai, információi mostantól nem az asztalon, a PC-je winchesterében lesznek elraktározva, hanem valahol a hálózaton. Hasonlóképp a vállalati internetes rendszerek esetében is ugyanez lesz később a helyzet.

Ennek a szemléletváltásnak hamarosan be kell következnie, amit a megbízhatóságot szavatoló, magas szintű technikai eljárások egyre inkább elősegítenek. Kialakul majd a felhasználók bizalma, ugyanúgy, mint ahogy az emberek régen megszokták: a pénzüket nem otthon, a nagypárna alatt őrzik, hanem megbízható bankokban tartják.

Mi mindent megteszünk azért, hogy az NC-technológia elterjedjen. Kidolgoztuk az NC-k hardver-referenciatervét, az NC operációs rendszert, a Network Computer Architecture-t, az NCA-t, az NC-s alkalmazásokat, és így tovább. Támogatjuk a gyártókat. Már sokféle prototípus ismert szerte a világon.

Az NC-technológia, illetve az Internet fejlődése kapcsán meg kell jegyeznem, hogy óriási jelentőségűnek tartom az egész világ jövője szempontjából az elektronikus kereskedelem elterjedését. Ezen a területen, azt hiszem, szintén nagy feladatokat vállal majd az Oracle.

Milyen új, aktuális technológiai fejlesztések hagyják el mostanában az Oracle laboratóriumait?

S. O.: Folytatva az internetes témakört: hamarosan valamennyi termékünknek, eszközünknek, alkalmazásunknak elkészül a webesített változata. Ez nagyon fontos, hiszen addig nem is lehetne áttérni PC-ről NC-re, amíg a felhasználók függenek a Windows-tól. A Windows-tól független megoldásra, felületre vagy interfészre van szükségük, ami lehetővé teszi, hogy bármivel rákapcsolódhassanak a hálózatra, akár NC-vel, akár PC-vel vagy Macintosh géppel. Teljesen mindegy, hogy a felhasználónak milyen gép van az asztalán, ha nem függ a Windows-tól. Ez alapvetően szükséges ahhoz, hogy az NC-korszakba léphessünk. A másik, amit mint technológiai újdonságot kiemelnék, az a néhány hónap múlva megjelenő Oracle 8 adatbázis-kezelő rendszer, melynek legfőbb jellemzője az erőteljes objektumorientáltság. Ez tulajdonképpen egy objektumrelációs adatbázis-kezelő, amellyel nagy teljesítményű, nagy megbízhatóságú, hatalmas adatbázisok hozhatók létre.

Arra kérjük, szóljon néhány szót az ön által vezetett cégről, az Oracle Hungary működéséről!

S. O.: Cégünknél nyolcvan munkatárs dolgozik. Bevételünk évről évre — a magyarországi informatikai piac bővülését meghaladóan — növekszik. Erre a pénzügyi évre nem kis célt tűztünk magunk elé: el kívánjuk érni a 2 milliárd forintos forgalmat. Ami a részarányos teljesítést illeti, a terv november végéig esedékes része beérkezett — mintegy 60 százalékos forgalomemelkedés a múlt év hasonló időszakához képest. Most dolgozunk a június elsejével induló, '98-as pénzügyi év bevételi tervein, és várható, hogy mintegy 2,5 milliárd forint lesz a tervezett forgalom.

Nagy projektjeink voltak az elmúlt időszakban, mind az Oracle Financials, mind az Oracle Manufacturing alkalmazást illetően. Megkezdtük a felkészülést, és a jövőben egyre többet foglalkozunk majd webes, internetes feladatokkal. November végén rendeztük meg az első magyarországi Network Computer bemutatót, ahol részletesen ismertettük az Oracle által kidolgozott Network Computing Architecture alkalmazásfejlesztő szoftverplatformot is. A téma iránti nagy hazai érdeklődést jól jelezte a rendezvényünkön megjelent sok száz szakember, üzletember aktív részvétele.



Oracle Hungary az Alkotás utcában



Nagy az érdeklődés az NC-technológia iránt

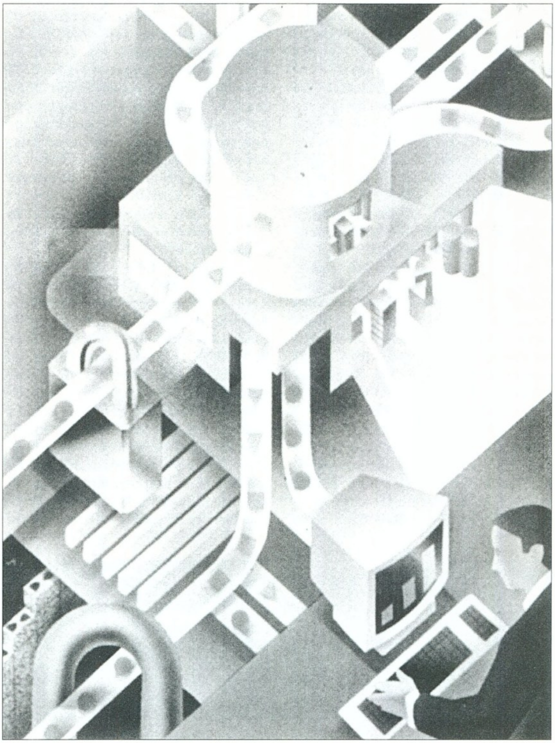
Vezetői döntéstámogató, Oracle Express termékcsaládunkkal kapcsolatban is jelentős árbevételt terveztünk, melyben fontos tételt képviselnek a néhány hónappal ezelőtti, komoly banki szállításaink. Oktatási tevékenységünk is sokat fejlődött az elmúlt hónapokban. Ma már 2400 felhasználót oktatunk különböző témakörökben egy év alatt. A legkorszerűbb műholdas technológiát is igénybe véve, az úgynevezett Oracle Channel útján Angliából sugárzott tanfolyamokat is szervezünk ügyfeleinknek. Hamarosan elindítjuk Interneten keresztüli kurzusainkat. Az Internet, a Web, az Oracle WebServer, a Java nyelv természetesen nemcsak eszköze, hanem témája is legújabb tanfolyamainknak.

Konzultációs tevékenységünk is gyorsan fejlődik, amire szükség is van, hiszen jelentős eredményeket szeretnénk elérni alkalmazási rendszereink értékesítésében. Ugyanakkor ambiciózus terveink megvalósításához együttműködő partnereket is keresünk; több céggel kötöttünk már ilyen célból megállapodásokat. A szoftverértékesítés hatvan százaléka jelenleg is együttműködő partnereken keresztül történik, akikkel továbbra is kiegyensúlyozott, eredményes együttműködésre törekszünk, és ehhez mindent megadunk, beleértve az egyre hatékonyabb oktatást, kiképzést is. ■

Data Warehousing Oracle Express Serverrel

A jó döntés alapja

A megalapozott vállalati döntések igazi biztosítékai lehetnek az Oracle Data Warehousing koncepció alapján létrejövő, a működési adatok kiértékelését végző vezetői információs rendszerek, melyek egyik legfontosabb eleme a multidimenzionális Oracle Express Server adatbázis-kezelő és a hozzá kapcsolódó eszközök. Az alábbiakban ezekkel foglalkozunk.



Az operatív, vállalati működési rendszerekben nap mint nap óriási mennyiségű adat keletkezik. Ezek kiegészülnek az úgynevezett történeti adatokkal is, amelyek a vállalatok korábbi időszakok alatti működését tükrözik. E nagy tömegű adatmennyiség komplex feldolgozását nevezzük Data Warehousing technológiának, illetve úgy is megfogalmazhatjuk, hogy a Data Warehousing azoknak a módszereknek és eszközöknek az összessége, amelyekkel nagy tömegű adatot lehet hatékonyan elemezni, és az analízis eredményeit jó hatásfokkal eljuttatni a felhasználókhoz.

Az Oracle Data Warehousing kezdeményezéséhez a cég relációs adatbázis-kezelőjében és más hagyományos termékeiben bevezetett újdonságok, új funkciók is hozzájárultak. A kezdeményezés legfontosabb eleme azonban az Oracle OLAP terméke, az Oracle Express Server multidimenzionális adatbázis-kezelő és a hozzá kapcsolódó, kliensoldali eszközök, amelyek lehetővé teszik a nagy tömegű adatok feldolgozását, analízisét és megfelelő interpretálását a felhasználók számára.

A szerver opcióként fontos része lett a nemrég megjelent Oracle Universal Server szuperterméknek. Az Oracle Universal Serverhez való opciós kapcsolódás azt is jelzi, hogy az adatok interpretálására az Oracle nemcsak az operatív rendszerekben megszokott eszközöket ajánlja, hanem egy új termékcsaládot, az Oracle Expresst is bevezeti.

Ezek az eszközök tulajdonképpen a frontendjei az Oracle Data Warehousing megoldásának, és ugyanakkor frontendjei az Oracle Applications Data Warehousing csoportnak is. Ez azt jelenti, hogy maguk az eszközök fel vannak készítve arra, hogy szervesen kapcsolódjanak az Oracle nagy alkalmazásaihoz és más, a piacon forgalomban lévő nagy alkalmazáscsomagokhoz az azokban tárolt adatok analízise céljából. Ugyanakkor ez nem jelenti azt, hogy a hagyományos relációs adatbázis-kezelőkön alapuló, testre szabott fejlesztések háttérbe szorulnának egy informatikai projekten belül, csupán arról van szó, hogy a fejlesztők egy újabb eszköztárat is igénybe vehetnek ahhoz, hogy még sokoldalúbban interpretálhassák a felhasználók számára az összegyűjtött adatokat.

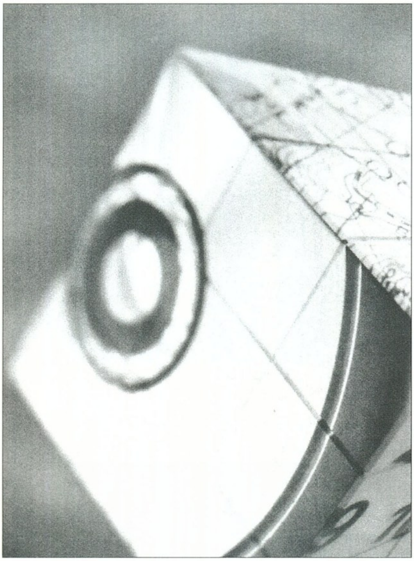
Az Oracle Express Servert és a hozzá kapcsolódó eszközöket a vállalati adatok lekérdezésére, kiértékelésére és optimális megjelenítésére fejlesztették ki. A multidimenzionális adatmodell és a megjelenítés eszköztára biztosítja, hogy a felhasználó vezetők a rendszeradminisztrátorok és a fejlesztők segítségének az igénybevétele nélkül is képesek legyenek hatékonyan elemezni az adatokat.

Az Oracle koncepciója szerint a Data Warehousing eszközök közé tartoznak az olyan, hagyományos fejlesztőeszközök is, mint az Oracle Developer 2000, valamint az adatbázisok ad hoc lekérdezését lehetővé tevő Discoverer család két tagja, az Oracle Data Query és az Oracle Browser, amelyeket úgy fejleszt tovább az Oracle, hogy azok szorosan integrálódhassanak az újonnan megjelenő kliensoldali OLAP eszközökkel. És természetesen a Data Ware-

housing fő vonalát jelenti az Oracle Express termékcsalád, az idetartozó Oracle Express Objects és Oracle Express Analyzer fejlesztőeszközök, továbbá a meglevő alkalmazások, például az Oracle Sales Analyzer és az Oracle Financial Analyzer.

„Az Oracle Express termékek olyan eszközök lehetnek a vállalati vezetők kezében, amelyekkel a vállalat működésekor nagy mennyiségben keletkezett adatokat oly hatékonyan képesek rövid idő alatt kiértékelni, hogy annak eredményeit előnyösen használhatják fel döntéseik meghozatalához. Például egy videók forgalmazásával foglalkozó cég vezetői pillanatok alatt képet kaphatnak arról, hogy az egyes videotípusok műszaki jellemzői vagy árviszonyai, esetleg az egyes árkedvezmény-akciók régiónként milyen mértékben befolyásolták az eladások számát" — mondja *Radnai Szabolcs*, az Oracle Hungary értékesítési konzultánsa. Az Oracle Express multidimenzionális adatbázis-kezelő termékcsalád olyan eszközök együttese, amellyel a komplex lekérdezések egyszerűsítése válik lehetővé. A termékcsalád középpontjában a multidimenzionális adatbázis-kezelő kernel (mag) áll, erre épülnek rá a család elemei. A multidimenzionális adatkezelés elve már korábban is ismert volt, de teljesen más alapokra épül, mint a megszokott relációs adatbázis-kezelés. Az adatokat úgynevezett dimenziók által kifeszített multidimenziós „kockákban" tárolja,

Az Oracle Express multidimenzionális adatbázis-kezelő termékcsalád olyan eszközök együttese, amellyel a komplex lekérdezések egyszerűsítése válik lehetővé.



A multidimenzionális adatbázis-kezelő által képzett mag köré szervesen épül fel a termékcsalád tagjait jelentő fejlesztőeszközkészlet.

ami nagyon hatékony lehet a többdimenziós lekérdezések megvalósítására. A multidimenzionális adatbázis-kezelőt a relációs adatbázisból vagy más, vállalati vagy külső adatforrásokból származó adatokkal kell feltölteni. Az Oracle Express termékcsalád nagy erőssége, hogy a különböző Express eszközökkel nemcsak a multidimenzionális adatbázis-kezelő segítségével tárolt adatok érhetők el, hanem lehetőség van az adatoknak közvetlenül a relációs adatbázis-kezelőből való kinyerésére is. Ez utóbbi történhet online módon, például ha egy analízis során különböző részletességgel kívánjuk elemezni forgalmazási adatainkat földrajzi koordináták szerint, eljuthatunk egy olyan részletességű szinthez, ahol már nem a multidimenzionális adatbázis-kezelőben, hanem a relációs adatbázis-kezelőben találhatók az adatok, ez esetben ide kell „lefúrni" (drill down) és felhozni azokat.

Az adatfeltöltés természetesen történhet nem online módon is: az adatokat ilyenkor batch jelleggel bizonyos időközönként tölthetjük be a multidimenzionális adatbázis-kezelőbe. A multidimenzionális adatbázis-kezelő által képzett mag köré szervesen épül fel a termékcsalád tagjait jelentő fejlesztőeszköz-készlet. E készletnek a közelmúltban készült el két nagyon fontos eleme. Az egyik az Oracle Express Objects, egy nagyon korszerű, objektumtechnológiával dolgozó fejlesztőegyüttes, ami lehetővé teszi a fejlesztő szakembereknek, hogy a korábbi projektekben elkészített objektumaikat új munkáik során is felhasználhassák, és így nagyon gyors fejlesztéseket végezhessenek. További fontos jellemzője az is, hogy működése könnyen elsajátítható, illetve egyszerű vele a fejlesztés. A készlet másik eleme a leszűkített, de hasonló funkciókat megvalósító Oracle Express Analyzer, ami az Oracle Express Objects-ben elkészült programok futtatására szolgál, mégpedig oly módon, hogy az adatok a felhasználók által nagyon könnyen kezelhető briefingekbe, azaz összefoglalókba vannak szervezve, melyek egyszerű navigálást tesznek lehetővé egy-egy témán belül.

A briefingek olyan, fejlesztő által létrehozott, úgynevezett lapok egységes csoportja, amelyeken lekérdezési objektumok, grafikonok, táblázatok helyezkednek el. A fejlesztő az Oracle Express Objects eszközzel létrehozza a megfelelő lekérdezéseket az adott szintű vezetők számára, ezeket a lekérdező oldalakat összefűzi, és úgy adja át (vagy például levelezőrendszere segítségével átküldi) a kiértékelést végző menedzsernek, hogy az nagyon egyszerűen navigálhasson az egyes lapok között. (A navigáláshoz a briefingek futtatásakor egy olyan „gombsor" button pad áll a menedzser rendelkezésére, amelyből az egyes gombok megnyomásával megkapja azokat a lapokat, amelyeket a fejlesztő előállított számára.)

„A fejlesztő által az Oracle Express Objects-ben előállított briefingek segítségével a felhasználó megfelelő értékeléseket, analíziseket végezhet el. Az elemzések lehetővé teszik a meghozott intézkedések hatékonyságának megítélését, sőt alkalmat adnak a további lépések hatásának előrejelzésére is. Nagyon lényeges tény, hogy ezekbe a termékekbe számos olyan elem van beépítve, ami elősegíti a helyes előrejelzések kimunkálását, amelyek például figyelembe veszik a szezonális változásokat, grafikonokat készítenek az egyes eladásokról, meghatározzák, hogy egy-egy termékből mennyit kell gyártani" — teszi hozzá Radnai Szabolcs.

Az Oracle Express termékcsalád kiegészül még a korszerű World Wide Web-elérést lehetővé tevő eszközökkel is. Ezek révén a vállalati felhasználók számára széles körben lehet publikálni az analízis-eredményeket. Így nem szükséges, hogy az egyszerűbb analízisekhez a vállalati felhasználók kliensoldali lekérdezővel, az Oracle Express Analyzer termékkel rendelkezzenek, elég, ha csupán egy, a kereskedelemben bárhol beszerezhető, olcsó Web browserrel kérdezik le az Oracle Express adatbázist. A vállalatok felhasználói széles körben hozzájuthatnak az információkhoz, méghozzá mindig a dinamikusan változó, aktuális adatokhoz.

Az Oracle Express termékcsalád tagjai sorában a fejlesztőeszközök mellett vannak alkalmazások is. Ezek közül a legnagyobb jelentőséggel két alkalmazási területet érdemes kiemelni: a pénzügyit és a kereskedelmit. A pénzügyi területre az Oracle az Oracle Financial Analyzer elnevezésű, a vállalatok pénzügyi rendszeréből áttekinthető adatkinyerést lehetővé tevő termékét ajánlja, amelyet már integrált is ismert pénzügyi alkalmazási rendszerével, az Oracle Financials szoftverrel. Ez azt jelenti, hogy az Oracle Financials alkalmazóinak nem kell programozási munkát végezniük az Oracle Financials rendszerből való kiértékelések elvégzéséhez. Kereskedelmi területre az Oracle Sales Analyzer elnevezésű terméket kínálják. Ezzel a termékkel olyan eszközkészlet kerül az értékelést végző szakemberek kezébe, amellyel egyszerűen értékelhetővé válnak a vállalatok legfőbb kereskedelmi mutatói. Az eszközökbe olyan algoritmusokat építettek be, amelyekkel megbecsülhető a vezetői döntések hatása a vállalat kereskedelmi eredményeire. ■

Az Oracle és az Internet

NC-korszak-alkotó

Az Oracle meghatározó szerepet játszik az Internet széles körű és megbízható alkalmazásának a biztosításában. Ennek része például az *Oracle WebServer* család megjelentetése, más termékeinek webesítése és az a kezdeményezés, amely a hálózati számítógép koncepciójának a kidolgozásához és az NC elfogadásához vezetett.



Az Oracle már korábban kijött második generációs, internetes megoldásokat magában foglaló kliens-szerver szuperrendszerével, a WebServerrel. Az Oracle WebServer család olyan információkezelést tesz lehetővé, melynek révén a World Wide Web a hálózati számítástechnikának az eddigieknél lényegesen hatékonyabb, integrált platformjává válhat. Eredményesen kombinálja a multimédiás Oracle7 adatbázis-technológia erejét a Web szerverekkel, a böngészőprogramokkal és a fejlesztőeszközökkel, támogatja az Interneten a nagy tömegű információszolgáltatást, az interaktív kommunikációt, az elektronikus kereskedelmet.

„A család fő eleme az *Oracle WebServer*, ami a korábban ismert szerverektől eltérően lehetővé teszi, hogy az adatbázisok élő, online módon kapcsolódhassanak a Webhez, és így az adatbázisban történő változások folyamatosan elérhetők legyenek az Interneten. Az előző megoldásokra jellemző volt, hogy ha az Internetre felvitt információban változás állt be, akkor az új verziót manuálisan el kellett készíteni és újra fel kellett vinni. Tehát az Oracle WebServer biztosította élő kapcsolat megjelenése igazi újdonság volt" — állítja *Klotz Tamás*, az Oracle Hungary értékesítési konzultánsa.

Az Oracle WebServer szoftver további fontos jellemzője, hogy bizonyos esetekben az Internetet igénybe vevő végfelhasználó nemcsak hozzájuthat a keresett információhoz, hanem maga is továbbíthat online módon információt az adatbázisba. Például nemcsak tájékozódhat az egyes repülőjáratok menetrendjéről, az esetleges késésekről, hanem azonnal meg is rendelheti a repülőjegyét a kiválasztott járatra.

Az új Oracle WebServer család másik fontos eleme az *Oracle PowerBrowser* böngészőprogram, amelyik két lényeges tulajdonságában is eltér az eddig ismert böngészőktől. Egyik ilyen jellemzője, hogy nemcsak az igényelt adatokat, hanem az azok felhasználásához valamiképp szükséges programokat is képes letölteni, például egy tömörített képpel együtt letölti a szükséges kitömörítő programot is.

Az Oracle jelentős eredményeket ért el az Internet-világban. Az első volt azon cégek sorában, amelyek teljes funkcionalitású szoftvertermékeket értékesítenek a Weben keresztül. Majd bemutatta az első „Csináld magad" eszközkészletet az adatbázisok és a Web összekötéséhez, amit azóta sok ezer felhasználó töltött le az Internetről.

Az Oracle WebServer család részeként megjelent Oracle Internet Commerce szoftver nagyon hatékony, olcsó eladási szolgáltatást kínál a Weben keresztül, az Oracle Store pedig könnyű és gyors megoldást nyújt az Oracle termékek WWW-n keresztüli értékesítéséhez.

„Elkészült és világszerte sikerrel használják az Oracle első internetes irodaautomatizálási rendszerét, az Oracle InterOffice alkalmazást, amely a magyar piacon is élénk érdeklődésre tarthat számot" — jelentette ki *Vadnai Tamás* értékesítési igazgató.

Ami az informatika szép új világát illeti, az Oracle kidolgozta a hálózati számítástechnika architektúráját (NCA), elkészítette a hálózati számítógép (NC) operációs rendszerét és hardver-referenciatervét, a világ nagy elektronikai cégeinél beindult az NC-k gyártása, elkészültek az első példányok. Megkezdődött a felkészülés a hamarosan beköszöntő NC-korszakra.

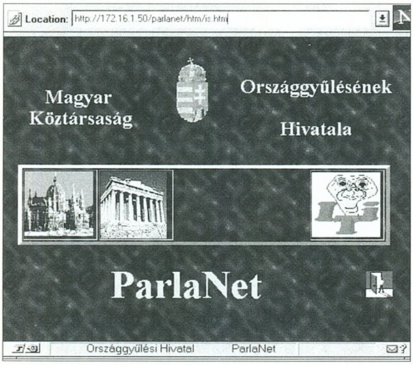
Az Oracle kialakította a nyílt platformokon átívelő, hálózatalapú architektúrát, a Network Computing Architecture-t (NCA), melyhez közzététele óta a világ nagy és kis szoftvercégei tömegesen csatlakoztak. Az NCA egy alkalmazásfejlesztési szoftvertechnológiai architektúra, amelyben az Oracle a kliens/szerver rendszerek robusztusságát, megbízhatóságát, az internetes, webes alkalmazások könnyű felhasználhatóságát és az objektumorientált világ elosztottságát, kiterjeszthetőségét egyesítette. Jellegzetessége, hogy a fejlesztőeszközök vagy például a vezetői információs rendszerek ún. kazettákként kapcsolódhatnak hozzá. „Az NCA közérthető, egyszerű módot kínál a kliens/szerver rendszerek integrálására az Internettel, az osztott objektumarchitektúrákkal, és így lehetőséget nyújt új típusú, nagyon hatékony, operatív, döntéstámogatási, csoportos munkát szolgáló, valamint elektronikus kereskedelmi megoldások kialakításához. Lehetővé teszi, hogy még hatékonyabb módon lehessen alkalmazásokat készíteni a Webre" — tette hozzá Klotz Tamás.

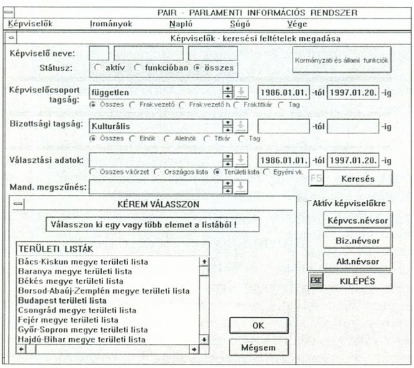
Az Oracle nagy szerepet tölt be a legújabb információtechnikai eszköz, a hálózati számítógép, az NC kidolgozásában és elterjesztésében. Az NC-k hardverfelépítésének referenciatervéből, az NC operációs rendszerből, valamint számos NC-s alkalmazásból álló technológiacsomagot egyetlen licenc keretében értékesítik a gyártó cégeknek. Ugyanakkor az Oracle különböző szolgáltatásokkal való koordinációjában is segíti a gyártókat. Nagynevű cégek bevonásával konzorciumot hozott létre, melynek feladata egyrészt annak garantálása, hogy a bármely cég által gyártott, de általa elfogadott NC megfelel a referenciatervben leírtaknak, másrészt a felülről való kompatibilitás szavatolása, azazhogy senkinek se kelljen lecserélnie meglévő NC-jét csak azért, mert egy újabb Web-alapú szolgáltatást kíván elérni. Az Oracle elsősorban azért vállalkozott az NC-technológia elindítására, mivel erőssége a szerverek biztosítása, amelyek révén kihasználhatók az NC-k millióiban rejlő lehetőségek.

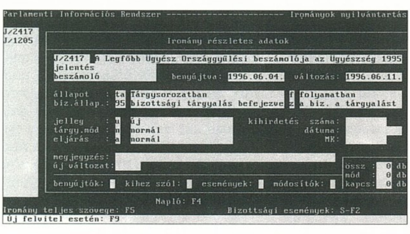
A ROVATOT ÍRTA ÉS SZERKESZTETTE:
CSÁNYI GYÖRGY

Indul a parlamenti intranet, a ParlaNet

A parlament munkáját kiszolgáló informatikai háttér „újkori" történelme tavaly januárban kezdődött — ekkortól működik a számítógép-hálózat Novell Netware 4.1-es operációs rendszerrel, mégpedig IPX-mentesen, tiszta TCP/IP architektúrával. *Dina István*, az országgyűlés informatikai osztályának vezetője szerint ez a megoldás egyrészt megfelel a parlamenti rendszer sajátosságainak, itt ugyanis minden kliensgépnek több Unix szerverrel kell kapcsolatban állnia, másrészt sok előnye van az IPX megoldásokkal szemben, mert így a hálózat ugyan több szempontból centralizált, mégis „demokratikusan" működtethető. Minden gépet regisztrálni kell, nem mozgathatók korlátlanul, vagyis a TCP/IP rendszer „rendet" követel. A hálózatra nyolc forrásból kapcsolódnak kliensgépek, a hét parlamenti frakció, valamint az informatikai fejlesztők gépei. Ez a rendszer arról is gondoskodik, hogy a szomszéd frakciók ne tévedhessenek egymás „vadászterületeire".







Jelenleg 650 PC kapcsolódik a parlamenti hálózatra. 1994 őszén állapodtak meg az országgyűlési frakciók egymás között arról, hogy mindegyikük kap alapellátásra öt számítógépet, s minden öt képviselő után egy-egy újabbat. A cél az, hogy mindenkire — a 386 képviselőre és a körülbelül 700 országgyűlési hivatali alkalmazottra — jusson egy-egy PC, azaz majdan hozzávetőleg 1000-1100 munkaállomás lesz a hálózatban.

A napjainkban működő szervergépek között van IBM és Hewlett-Packard is, a PC-k között találhatók 286-osok és 386-osok is, a gépek nagyjából 50 százaléka olyan kiépítettségű, hogy jól megy rajta a Windows. A parlament informatikai rendszerét egy 25 főből álló csapat fejleszti, üzemelteti.

Dina István tapasztalatai szerint a képviselőknek általában 10 százaléka használ úgymond „direktben" számítógépet, nemcsak Magyarországon, de a világ más parlamentjeiben is. A többieknek az alkalmazottai, szakértői dolgoznak a gépekkel.

A parlamenti információs rendszer által nyújtott szolgáltatások három csoportba sorolhatók. Az elsőbe tartoznak az irodaautomatizálási rendszerek, a Microsoft Office, az Excel, de a PowerPointot is használják néhányan.

Másik szolgáltatási csoport az adatbázis-hozzáféréseké. Oracle-alapú, karakteres adatbázisokat építettek fel, ezek tartalmazzák a képviselőkre és a szavazásokra vonatkozó információkat, az iratanyagokat és a felszólalások szövegét, jelenleg mintegy 400 megabájtnyi adatot. Mindezeket egy 70 pontból álló gráfhálózatba szervezték a parlamenti házszabály kívülállóknak meglehetősen bonyolult állapotai szerint, azaz a rendszer tudja, hogy ki milyen anyagot nyújthat be, mikor milyen állapotban van egy dokumentum stb.

Az egyes országok parlamentjei egyébként igencsak eltérnek egymástól abban, hogy mit őriznek meg és milyen formában az ülésekről, vitákról. Van, ahol a hangdokumentumokat tárolják akusztikus fájlokban, másutt videovágóképeket készítenek, és így tovább.

A magyar parlamentnek is vannak sajátosságai, hogy mást ne említsünk, az itteni információs rendszernek fontos feladata megőrizni minden képviselő minden egyedi szavazását, mint ahogy jelentősek a legkülönbözőbb politikai statisztikák is, úgymint hányszor szavazott egyformán az MSZP- és az SZDSZ-frakció (sokszor) stb. Külső adatbázisok anyagához is hozzáférhetnek a képviselők a kliensgépekről, ezek közé tartoznak például a céghírek, a jogtár, a telefonkönyv és néhány szótár.

A szolgáltatások újabb csoportja a levelezés: az E-mail két éve használható, jól működik, sokak által kedvelt. A parlamenti Web-lapokról az Infopen már beszámolt a Rulez-díj odaítélése kapcsán *(1996. aug.-szept. szám)*.

A februári ülésszakkal egy időben indul — hamar követve a világtendenciát — a legújabb szolgáltatás, a parlamenti intranet, ami a ParlaNet nevet kapta a keresztségben. Ehhez nem kellenek bonyolult bejelentkezési jogosítványok, igazán egyszerűen tájékozódhat mindenki. Tartalmaz minden gépi formában rögzített országgyűlési információt, s fejlesztésekor ügyeltek arra is, hogy többféle „horgot", azaz kíváncsivá tevő elemeket is elhelyezzenek benne, így például szabadidős ötleteket is ad a rendszer, kapcsolatokat az Interneten lévő sajtótermékek címlapjaihoz.

TAKÁCS GITTA



http://www.informix.com

Információgyűjtés és osztályozás, feldolgozás és tárolás, előkészítés és lekérdezés - embereket és vállalatokat egyaránt próbáratevő összetett és bonyolult folyamat. Az Informix a keresett információ zökkenőmentes elérését teszi lehetővé a megfelelő időben, a megfelelő helyen, megfelelően feldolgozva.

Bár a feladat hétköznapi, megvalósítása kimagaslóan igényes az információgazdálkodás területén.

Az INFORMIX® az információ univerzális hozzáférését biztosítja az Önök igényeihez igazítva.

Dinamikus, objektum-relációs adatbázis technológiánk - mely bármilyen típusú információ kezelésére alkalmas - a vállalati szintű információs infrastruktúra alapja.
Stratégiánkat 15 éve fennálló technológiai vezetőszerepünk is igazolja.

INFORMIX®
Technology Center Hungary

InTeC Hungary Kft.
1063 Budapest, Bajnok u. 13., Tel: (06-1) 302 3388, Fax: (06-1) 302 3395

Hogyan készítsünk saját honlapot a Weben?

NIIF FÜZETEK II/10
Szerző: Perlaki Attila
(irtpa@gold.uni-miskolc.hu)
http://www.lib.uni-miskolc.hu/ali
Kiadó:
Nemzeti Információs Infrastruktúra
Fejlesztési Program
1996. szeptember
A kiadvány elérhető a
http://www.iif.hu/dokumentumok címen

Töltsük be számítógépünkre a Netscape vagy az Internet Explorer (avagy más meglevő) netsurfer programunkat, amit kissé szabadon, de értelmének megfelelően böngészőnek fogunk nevezni. Bökjük meg az Open gombot, és írjuk be az ablakba: *http://www.lib.uni-miskolc.hu/ali*. Betöltődött? Nos, ez egy home page, azaz honlap. Egy lap a több mint tízmillióból az Interneten. Ez a füzet azoknak készült, akik növelni szeretnék ezt a számot a saját honlapjukkal.

Honlapot intézmények és magánszemélyek egyaránt készítenek. Előbbiek prospektus és reklám mellett vagy helyett, illetve valamilyen szolgáltatást nyújtva (pl. keresőrendszerek, adattárak); utóbbiak önmaguk vagy kedvenc dilijük bemutatása céljából. Van honlapja a Földnek, Magyarországnak, újságoknak, focicsapatoknak, a Télapónak és számos rajzfilmhősnek. A honlap a szigorú hivatalosságtól a korlátlan őrületig mindent tartalmazhat — egyfajta önmegvalósítás. Hé, világ! Itt a honlapom, nézd! És a világ megnézi.

Először a CERN-ben dolgozták ki a honlapot és az elkészítéséhez szükséges segédeszközöket. Miért pont az atomtudósok környékén született meg ez a példátlanul népszerű és megállíthatatlanul terjedő eszköz? Ez egy kicsit a véletlen műve.

Valamikor az ősidőkben (vagyis pár éve — a számítástechnikában ez nagy idő) már megalkották a Gophert, mint hatékony és könnyen kezelhető információs rendszert. Végre elég volt csupán pár billentyű kezelését megtanulni ahhoz, hogy az ember a világ bármely pontjáról — áttekinthetően és gyorsan — információhoz jusson. Ekkor már az Internet egységes egészként jelent meg a felhasználó előtt. Az, hogy a kért információ milyen gépen, milyen operációs rendszer alatt és a világ melyik pontján van (az átviteli sebességet leszámítva), érdektelenné vált. Gondoljuk meg, mekkora jelentősége van ennek — az egész világot „átláthatjuk", nincsenek többé határok!

A CERN tudósai éppoly lusták és válogatósak lehettek, mint bárki más, csak sokkal igényesebbek. A Gophert annak idején Minnesota egyetemén karakteres képernyőkre álmodták meg, és csak „unalmas" menüket tartalmazott. Az atomfizikusok meg ábrákat és képeket követeltek, továbbá dőlt betűt, s mindenekfelett szívükhöz nőtt a hipertext. Megkapták.

A honlapok a számukra kidolgozott kommunikációs szabványok szerint „közlekednek" az Interneten, és ezt értik a World Wide Web böngészőprogramok. A lapok szerkesztő- és kiegészítőjelekkel (leírókkal) ellátott szövegállományok. A leírók szabványa a HTML (HyperText Markup Language), amely egyfajta leírónyelv, a lap megjelenését tudjuk „programozni" vele. Jelenleg a 3.2-es verziónál tart.

A célom az, hogy aki e sorokat olvassa, önállóan is bele merjen vágni a honlapkészítésbe — ez nem ördöngösség! Az alapok elsajátítása után pedig egyet tudok ajánlani: a View/Source gomb szorgos nyomogatását, ha érdekes ötlettel találkozunk barangolásaink során! A HTML lapokban talált megoldások ugyanis közkincsnek számítanak, a bennük fellelt trükkök szabadon felhasználhatók (a tartalom viszont nem biztos). El is menthetjük a File/Save as/Source kiválasztásával, s ebből tanulhatunk tovább.

Mi kell a honlaphoz?

Ha nekifogunk a honlap készítésének, legelőször keresnünk kell egy gépet, amin majd dolgozni fogunk. A lap építéséhez megfelel egy „mezei" PC is, kipróbálásához már egy linuxos vagy Windows-os Netscape, vagy hasonló böngészőprogram szükséges. Amennyiben a lapunkat őrző géppel csak rövid ideig tudunk dolgozni, pl. azért, mert telefonon át tartjuk rendszeresen a kapcsolatot, akkor érdemesebb az otthoni gépen megírni és kipróbálni a lapo(ka)t, majd a kész változatot felküldeni. Ahhoz ugyanis, hogy mások is megtekinthessék a lapunkat, *Internet kapcsolat* szükséges. Ezt minden guru host néven emlegeti; ami annyit tesz, mint „házigazda", s valóban, nála lakik majd a lapunk. Ennek a gépnek az Interneten ismert címe, sőt szinte mindig neve is van. Képzeletbeli hostunk hallgasson a *www.honlapiskola.hu* névre. Ez a gép tehát rajta van a Neten, méghozzá a nap mind a huszonnégy órájában. Kerüljük el a korlátozott nyitva tartással üzemelő hostokat, mert a Net az egész bolygót magában foglalja, és mindig ébren van. Szerény lapunkat pedig Tokiótól Washingtonig bárhonnan meglátogathatják, mert nincsenek távolságok ebben a furcsa világban. Ahhoz, hogy dolgozni tudjunk, be kell jutnunk a hostra. Számos host biztosít anonim, nyilvános hozzáférést bizonyos szolgáltatások használatára. Ezzel akár az Internet bármely szegletébe is elvitorlázhatunk, saját lap alapításához azonban nem elég. Kell egy *Account* is, amit postafióknak, hozzáférésnek is próbálnak fordítani, de nem egészen találó. Lényegében arról van szó, hogy a host személyesen ismerjen minket, és kínáljon fel számunkra egy zugot, ahová virtuális ingóságainkkal beköltözhetünk.

Az accounttal együtt jár egy usernév és egy password is. (A password mindig nehezen kitalálható legyen, és legalább havonta változtassuk! A hackerek sohasem alszanak!)

Tegyük fel, hogy nekünk a Jakab név jutott. Jelentkezzünk be! Valami hasonlót látunk majd:

```
Welcome at www.honlapiskola.hu
login: jakab
password: ********
```

Ha ezek után a gép morc módon ki nem dobott minket, akkor valóban van élő accountunk. Ide akár levelet is kaphatunk, jegyezzük is meg a címünket:

`jakab@www.honlapiskola.hu`

Kérjünk meg egy ismerőst, aki nem erre a gépre kapott accountot, hogy küldjön a címünkre levelet. Ha megkapjuk, a host Internet-elérése rendben van. Ezenkívül futnia kell egy *httpd* programnak is. Ezt a host korlátlan ura és parancsolója, a root (rendszergazda) helyezi üzembe (azaz mi ezzel ne próbálkozzunk!), és ha nem értünk igazán a host operációs rendszeréhez, akkor a legegyszerűbb ellenőrzési mód a host kezdőlapjának hívása. Az általunk használt böngészőbe írjuk be új címként:

`http://www.honlapiskola.hu`

Ha erre megjelenik valami szépséges oldal, nyert ügyünk van (majdnem). Ha nem, akkor érdeklődjünk a hoston más userektől, végső esetben a roottól, valaki csak fogja tudni, miért nem működik... Lehet,

hogy a httpd program nem fut, de az is előfordulhat, hogy nincs kezdőlap (ez elég szomorú) vagy más címen van (nem „www" a host neve, vagy nem a szokásos „index.html" a kezdőlap címe).

Ideje csinálni valamit, ha már bejelentkeztünk! Hozzuk létre azt a helyet, ahova a lap kerül. Ez a *public_html könyvtár,* vagy directory, vagy folder, kinek ahogy tetszik. Egy parancs az egész:

```
mkdir public_html
```

Kész a „telek", kezdhetjük építeni a „házat". Először is lépjünk be a „telekre":

```
cd public_html
```

Ha bent vagyunk, akkor a ház bejárata az index.html állomány. Egyelőre legyen benne csak annyi, hogy „Under construction!", azaz magyarul „Fejlesztés alatt!" Lássuk!

```
echo „Under construction!" >
index.html
```

Nocsak, már készen is van az első lapunk, vagyis inkább a nulladik. Ellenőrizzük a hozzáférési jogot (ls-al), s ha megvan a megfelelő számú r és x, akkor a lap címe az Interneten:

```
http://www.honlapiskola.hu/~jakab
```

Az első lap

Készítésénél nem kell ágyúval verébre lőni, eleinte bármilyen egyszerű karakteres szövegszerkesztő megteszi. UNIX alatt a pico vagy a joe (a vi az elborult elmék varázseditora, igazi buherátor nem is nyúl máshoz, de kezdőknek nem ajánlom); DOS alatt a Norton Editor vagy az EDIT is megfelel, illetve Windows alatt a Notepad.

Tehát elkezdjük a bejárat, azaz az index.html szerkesztését.

Egy HTML lap szerkezete a HTML „nyelv" szabályai szerint így néz ki:

```
<html>
<head>
Fej
</head>
<body>
Törzs
</body>
</html>
```

Mi kerül a fejbe?

A fejben a lapra jellemző, rövid, pár szavas leírást kell megadnunk. Ez a szöveg a böngészőprogram ablakának tetején fog megjelenni, illetve a hálózaton üzemelő keresőautomaták is ezt írják majd a lap hálózati címe mellé. Meg kell tehát gondolni, mi is kerül ide. Ráadásul ékezetet sem ajánlott használni, mert nem mindegyik böngésző- és keresőprogram képes azt helyesen értelmezni.

Mi kerül a törzsbe?

Maga a lap, amit írunk. Az összes szöveg, a képek helye, miegyéb.

Ezek után Jakab számára az első kész lap valahogy így néz ki:

```
<html>
<head>
<title>Jakab honlapja</title>
</head>
<body>
<pre>
Hello!
Jakab vagyok, most tanulom a
honlapkészítés rejtelmeit. Hamaro-
san ez az oldal tele lesz minden
széppel és jóval.
A levélcímem:
jakab@www.honlapiskola.hu
</pre>
</body>
</html>
```

Nos, ezzel valóban kész az első lapunk. A szöveg szabadon választott, lehet játszani. De hamar felmerül a kérdés:

Mi kerüljön az első lapra?

Amikor az ember honlapot készít, nyilván igyekszik bemutatni magát. Ezt számtalan módon meg lehet tenni, ez itt a korlátlan önmegvalósítás ideális terepe. Szokásos megoldás fényképünket és néhány alapvető adatot kirakni, de lehet humoros kisregényt írni magunkról fénykép nélkül, vagy szigorú szakmai önéletrajzot három nyelven, vagy játékos ákombákomokkal teleaggatni az egész lapot. Mindent lehet. Egy dolgot érdemes szem előtt tartani: a lap információs erejét. Érdektelen, alig áttekinthető, nehezen letöltődő (azaz nagyon hosszú) lapokról hamar tovalebben az Interneten bolyongó!

Lehet tiszta lappal kezdeni a hálózaton, de ha már egyszer valamit valamikor leírtunk, nem szívesen pötyögjük be újra. Azaz előbb-utóbb elkövetkezik a kész szövegek bevágása a lapba. A legegyszerűbb, ha van egy „primitív" szövegszerkesztővel elkészített, általában .txt végződést viselő állományunk. Ha nincs benne ékezet (mert mondjuk angol nyelvű), akkor még jobb. Nincs más dolgunk, mint a szöveg elejére beírni: <pre> és a végére: </pre>. Ezt a törzsbe beillesztjük, és kész.

Olcsó húsnak híg a leve: a preformatted (előre tördelt) szövegek csúnyácskák, és gyanúsan úgy néznek ki, mint egy program forráslistája egy monitoron. Nem csoda, ezt a megoldást pont erre találták ki! A böngészőprogramok ugyanis egyébként figyelmen kívül hagyják az általunk beírt sorvégjeleket és tördelést, öntik a szöveget, ahogy az az ablakba befér. Próbáljuk ki! Egy hosszabb .txt állományt először zárjunk <pre> és </pre> jelek közé, s nézzük meg:

```
<pre>
Hol az a dal,
Ami felkavar?
Hol az a dal,
Ami vigasztal?
Hol az a dal,
Ami egyszer észrevétlenül elve-
szett?
</pre>
```

... majd töröljük őket, és gyönyörködjünk: Hol az a dal, Ami felkavar? Hol az a dal, Ami vigasztal? Hol az a dal, Ami egyszer észrevétlenül elveszett?

Ugye milyen szép? Jó, hogy láttuk, mert később, ha valamiért olvashatatlan betűfolyammá válik a lapunk, gyanakodhatunk arra, hogy valahol valamilyen HTML jelet elfelejtettünk!

Az előre tördelt szövegekkel más baj is van: a böngésző nem mer hozzájuk nyúlni, azaz ha hosszabb a sor, mint amilyen széles az ablak, akkor nem látjuk az egészet — tologathatjuk a görgetősávot oda-vissza. Korlátozhatjuk persze a szélességét, de valaki bizonyosan annál is kisebb ablakot használ, mint amit mi feltételezünk...

Ha viszont programok forráslistáját kell közölnünk, nehogy mást használjunk! Nincs reménytelenebb meló, mint egy precíz bekezdésekkel ellátott forrásszöveget újraszerkeszteni. A böngészők, hogy kihangsúlyozzák, itt valami másról van szó, s nem közönséges szövegről, az előre tördelt részeket más, írógépszerű betűtípussal jelenítik meg.

Nem lebecsülésként írtam az elején azt, hogy „primitív" szövegszerkesztő. Ma ugyanis a WinWord is szövegszerkesztő programnak számít. „Primitív" az a szövegszerkesztő, amelyik bárhol olvasható, szerkesztőjelek nélküli, ún. plain text állományt állít elő. Azaz az állományban csak a szöveg van, semmi más. Nevezhetjük nyers szövegnek is. Amennyiben nem nyers a szöveg, hanem már szépen tördelt valamelyik nagytudású programmal, akkor lépnek színre a *konverterek.*

Annyiféle formátum van már, hogy az elképesztő. Reménytelen vállalkozás tehát univerzális átalakítót keresni vagy írni. Léteznek eredményes kísérletek olyan állományformátumokra, amelyek az összes elképzelhető szerkesztési lehetőségre fel vannak készítve; ilyen például a Rich Text Format (.rtf). Figyelem, ez a formátum közönséges halandó számára olvashatatlan! Olyan, mint egy lefordított program, csak itt a vastag betű, dőlt betű, ilyen betű, olyan betű van egységes „nyelvre" fordítva. Ezt a formátumot a szövegszerkesztő programunknak ismernie illik! (A WinWord például ismeri.)

A már megszerkesztett szöveget tehát ebben a formátumban újra el kell mentenünk. Utána pedig keresni kell egy olyan programot, amelyik .rtf állományból .htm(l) állományt készít. Ilyen az rtftohtm nevű, FTP archívumokból letölthető shareware program. Használata egyszerű:

```
rtftohtm valami.rtf
```

A program elfogadhatóan dolgozik, amíg valami extra nem bukkan fel a szövegben (pl. egy szöveggel körülöntött ábra), ezen biztosan elbukik majd. Bevallom, ez nem is baj. A HTML leírások ugyanis csak támpontokat adnak a megjelenítésre, és

nem definiálják azt szigorúan. Azaz: a Web lap megjelenése a böngésző beállításától is függ, semmi garancia nincs arra, hogy más is azt látja, amit mi!

A Web lapon nem lehet cél egy profi képesújság mérethelyes megvalósítása. Kész szövegeink tartalma a lényeg, a formát pedig igazítanunk kell a lehetőségekhez, esetleg *kézzel szerkesztve.*

Ha üres lappal indítunk, vagy ha már kész anyagot kell Webre hangolni, akkor a megfelelő szerkesztési „utasításokat" magunk is elhelyezhetjük.

Láttuk már, hogy ezeket az „utasításokat" < és > jelek közé zártuk, ez minden esetben így van. Leírásukhoz kis- és nagybetű egyaránt megfelel, a kisbetű talán kényelmesebb, a nagybetű jobban látszik. Vannak jelek, amik párban állnak (zárni kell őket), a bezáró egy / jellel többet tartalmaz.

A szöveg bekezdéseinek elhatárolásához a <p> jelet használjuk. A bekezdéseken belül a szöveget a betűméret és az ablakszélesség figyelembevételével szabadon tördeli a böngészőprogram, ám ennél a jelnél mindenképpen új sort kezd, a két bekezdés közt általában üres sort is hagy. Ha szükséges, külön is kérhetjük az új sor kezdését (pl. párbeszédeknél) a
 jellel. Ez nem hagy üres sort. Ha a szövegben élesebb elhatárolást szeretnénk, úgy vonalat is húzhatunk a <hr> jellel.

Egy hosszabb szövegben valószínűleg fejezet és alfejezetcímek is előfordulnak, ezeket a <h1>, <h2>, ..., <h6> jelekkel emelhetjük ki. Figyelem, ezt a jelzést zárni kell! A <h1> párja a cím végén a </h1>. Az egyes a legnagyobb méretű betűt adja, a hatos a legkisebbet. Maga a szöveg a négyessel egyező méretű, így az ötös és hatos „címleírót" inkább speciális célra (pl. copyright jelzésére) használjuk.

Az amerikaiak balra igazítják a címeket, ha ez nem tetszik, a <center> jellel kérhetjük a középre igazítást. Ezt a jelzést is zárni kell! Összevonható, sőt illik is összevonni a címjelzővel, ekkor így alakul: <h1 align=center> és értelemszerűen elég csak így zárni: </h1>.

A szövegen belül vastag és dőlt betűs kiemeléssel élhetünk, erre a <b> és az <i> jelek szolgálnak, és ezeket is zárni kell. Ez a két jelző azonban egyes vélemények szerint kerülendő, ugyanis a HTML filozófiájának ellentmondanak: szigorú utasítást adnak, fizikai leírók és nem „útmutatók". (Valóban, van olyan eset, hogy nincs mód pl. dőlt betűt írni.) Ezért helyettük más, a logikai jelentést jobban leíró jeleket használhatunk.

Megerősíthetjük mondanivalónkat a <strong> jelzővel, hangsúlyt adhatunk neki az <em> jelzővel, vagy idézhetünk is a <cite> jelzővel. Levélcímünket illik <address> jelzők közé zárni — azaz ez is bezárandó, mint a többi.

Lássuk, Jakab lapja hogyan néz ki, feltéve, hogy szereti Koncz Zsuzsát hallgatni:

```
<html>
<head>
<title>Jakab honlapja V2.0</title>
</head>
<body>
Hello Mindenki!<p>
Ez itt
<h1 align=center>Jakab
honlapja</h1>
Jakab vagyok, most tanulom a
honlapkészítés rejtelmeit. Hamaro-
san ez az oldal tele lesz minden
széppel és jóval. <p>
A <strong>kedvencem</strong> sorait
idézem:
<p>
<cite>
Ma már a szívünk helyén kompjuter
dobol,<br>
Kiszámított a szenvedély.<br>
Egy új világ helyett csak látványt
tervezünk.<br>
A lelkünk hallgat, s csak a pénz
beszél.
</cite>
<hr>
A levélcímem:<br>
<address>
jakab@www.honlapiskola.hu<br>
</address>
</body>
</html>
```

Végül, mert sajnos idetartozik, néhány szó az *átviteli sebességről.* Nincs az a vonal, ami előbb-utóbb el ne dugulna a forgalomtól. Ekkor pedig egy nagy képet letölteni idegölő dolog. A fejezet elején említett mondás megfordul: ezer szó felér egy képpel, ráadásul gyorsabban átjön a hálózaton.

Kíméljük egymást! Legyen csak annyi képünk, amennyi feltétlenül szükséges (azért a lapunk egyéni arculatát nem kell agyoncsapni), és ezek legalább tárméretben kicsik legyenek. Ha más megoldás nincs, szedjük szét több lapra az eredetit. Egy lapra 40-60 kbájtnál több kép csak igen-igen indokolt esetben kerüljön. Számítsunk arra is, hogy (főleg a tapasztalt barangolók) kikapcsolják a képek letöltését — azaz a lap megtervezésekor fordítsunk gondot arra, hogy képek nélkül is mutasson valahogy. Számos eset van, amikor ez megoldhatatlan, de amikor van mód rá, legalább akkor tegyük meg. Vagy az alt= segítségével mellékeljünk magyarázatot a képekhez, vagy készítsünk önálló „text only" verziót is a sávszélességben „szegények" számára (ez gyakori megoldás).

Ha valamilyen okból sok kép kell a lapra, akkor is van mód a takarékosságra. Ne magukat a képeket tegyük fel, hanem kicsiny ikonokat, s ezek mögé pakoljuk be az egész estét betöltő plakátképeinket.

Merre tovább?

A HTML fejlődése ellentmondásos. Népszerűsége óriási, alapjai biztosak, jól átgondoltak, de azóta mindenféle torzulások keletkeztek benne. A baj az, hogy eredetileg egy nagyon egyszerű dokumentumleíró nyelvet definiáltak, most pedig arra tart a HTML, hogy GUI, azaz grafikus felhasználói felület lesz belőle (plusz multimédia alkalmazás és reklámhordozó, plusz DTP kiadványszerkesztő és -terjesztő eszköz, plusz belső vállalati információs rendszer és üzleti tranzakciók közvetítője stb.). Ma már rendelhetünk és fizethetünk is a WWW-ben bolyongva, laborokba kukkanthatunk be kandi kamerával, rádiót hallgathatunk egyenes adásban, és mindenféle adatbázis-lekérdezőket nyüstölhetünk kérdéseinkkel. Messze kerültünk a kezdetektől, és alaposan meghaladtuk a svájci atomtudósok igényeit.

Honlap holnap

Honlapunk címét elektronikus leveleinkben (a .signature állományban) és akár névjegykártyánkon is feltüntethetjük. Bejegyeztethetjük a Hungary Network „Ki-Kicsoda" lapjára *(http://www.hungary.com/kiki/)* vagy más szolgáltatókhoz. Más keresőszolgáltatások viszont hamarosan maguktól is rátalálnak a lapunkra. A látogatók száma eleinte kevés lesz, de ha jó az anyagunk, hamarosan egymás sarkát tapossák majd.

A jó lap nem avul el, azaz illik rendszeresen a karbantartással foglalkozni. Az elavult anyagokat és linkeket törölni kell, az újdonságokat és a még fejlesztés alatt álló oldalakat pedig külön érdemes jelezni, hogy a visszajáró látogatók könnyebben megtalálják. Ha véletlenül költözünk, akkor a régi címen (ha lehetséges) hagyjunk egy lapot. Ezen jó nagy betűkkel jelentsük be, hogy elköltöztünk, az új címünk pedig ez és ez, amit persze kattinthatóan építünk be.

Van egy-két dolog, amit nem szerencsés az oldalakra feltenni. A kimondottan reklámra szakosodott szolgáltatókon kívül a reklám általában tiltott tevékenység. Nem illik másokat megsérteni, bosszantani, és politizálni sem közkedvelt dolog. Ne tegyünk ki olyan anyagokat sem, amit a törvény tilt vagy az etikett. Ezeket egy kirakatba sem raknánk ki, és a honlap nem más, mint egy elektronikus kirakat. A rendszergazdától érdeklődjünk, mit tekint ő nemkívánatos anyagnak.

Mindezek nem lehetnek túl szigorú megkötések a végtelen fantázia számára. S lám, a megvalósítás nem is olyan bonyolult. Nincs mese, bele kell vágni a lapalapításba! ■

Az objektumorientált technológia sikerének titka

Mi gerjeszti a szoftverelőállítási módszerek változását? Erre a kérdésre egyszerű a válasz: az elkészítendő programok mindinkább komplex feladatokat oldanak meg. A hardvergyártók közötti verseny következtében a szoftvereket futtató „vas" egyre nagyobb teljesítményű és egyre olcsóbb. Emiatt mind szélesebb körben igénylik a műszaki üzleti élet feladatainak szoftveres támogatását. Az alábbiakban az IQSOFT szoftverfejlesztési projektmenedzsere osztja meg olvasóinkkal — főleg informatikai vezetőknek szánt — gondolatait az OO technológiáról.

A néhány évvel ezelőtti karakteres képernyők és gyakran kötegelt (batch) feldolgozást támogató programok helyett ma már grafikus felhasználói felületet, közvetlen (online) adatkezelést, az irodaautomatizálási szoftverekkel (szövegszerkesztő, táblázatkezelő, határidőnapló [schedule], fax, elektronikus levelezés [E-mail] stb.) való együttműködést (integráció) várunk el egy jó minőségű üzleti szoftvertől. Az előállítandó szoftverek bonyolultsága (komplexitása) oly mértékben megnövekedett (és folyamatosan növekszik), hogy a megbízható működés érdekében a szoftverfejlesztőknek az új igényekhez kellett igazítaniuk a fejlesztési (tervezési és programozási) módszereket. Már nem lehet hatékonyan elkészíteni az 1980-as évek (vagy még régebbi időszak) módszereivel a nagy bonyolultságú szoftvereket. A relációs adatbázis-kezelők (IBM DB2, Oracle, Informix, Sybase stb.) már a '80-as években előtérbe kerültek, de napjainkra érték el azt a fejlettségi szintet és elterjedtséget, ami az említett komplex szoftverekhez szükséges. A '90-es évek elején sorra megjelentek az interaktív (vizuális), grafikus képernyő- és riporttervezést támogató, integrált, 4 generációs (4GL) eszközök. A kliens-szerver (ügyfél-kiszolgáló) architektúrát megvalósító rendszerekben a szerver egy relációs adatbázis-kezelő, a kliens pedig PC-ken működő 4GL-ben készült célprogram, amely a szerver (adatbázis-kezelő) szolgáltatásait hívja.

Komponensalapú fejlesztés

Ehhez a „felálláshoz" fejlődtek ki a strukturált tervezési (SSADM, ORACLE*CASE stb.) és programozási módszerek, s érték el virágkorukat 1994-95-re. Először a kliensoldalon merült fel az igény — mivel ezen a területen volt a legfokozottabb ütemű a fejlődés — egy még nagyobb rugalmasságot és hatékonyságot biztosító fejlesztési módszerre. A strukturált módszerre alapuló 4GL eszközök csak a gyártó által meghatározott feladatok elvégzésére voltak hatékonyan alkalmazhatóak.

Ez idő tájt tört előre a komponensalapú fejlesztés is, ami a programozási nyelvtől, 4GL-től független, binárisan (gépikód szinten) kompatíbilis objektumok létrehozását és az ezekből történő rendszerépítést támogatja. A gyakorlatban először a Microsoft Visual Basicben kifejlesztett VBX (Visual Basic Extension) szabványra alapuló komponensek képviselték. Gyártásuk egész iparággá nőtte ki magát; százával (ezrével) készültek a VBX komponensek. Binárisan kompatíbilisek voltak (azáltal, hogy C-ben készültek) mindazon fejlesztőeszközökkel, amelyek támogatták a VBX (bináris) szabványt. Az OO4GL-ek gyártói is sorra VBX támogatást építettek termékeikbe: Gupta SQLWindows, PowerSoft PowerBuilder, Borland Delphi, Platinum ObjectPro stb.

A VBX-ek problémája: csak 16 bites Microsoft Windows környezetben működnek, azaz nincs VBX támogatás a Unix, OS/2, Mac világában, és maga a Microsoft sem támogatja a Win32 alkalmazásokban. A VBX helyett találták ki a Microsoftnál az OCX-et (mai nevén ActiveX-et), amelyet 16 bites és 32 bites Windows alkalmazásokban egyaránt lehet alkalmazni.

A Java nyelv megjelenésével és robbanásszerű sikerével teljesen egyértelművé vált az objektumorientált „tökéletes" programnyelv, ami szinte bármilyen szoftverfeladat megoldására alkalmas, teljesen hordozható, szabványos objektumkönyvtárakkal van ellátva, s a nagy gyártók (IBM, Microsoft, Oracle stb.) támogatását is élvezi.

A komponensek készítésének és kommunikációjának (protokolljának) szabályait a Microsoft COM (Component Object Model) és DCOM (Distributed COM) specifikációjában fektette le. COM és DCOM objektumok azonban csak a Microsoft Windows világban létezhetnek. Ha valaki általánosabb komponens-architektúrára szeretné alapozni szoftverrendszereit, akkor az Objekt Management Group (OMG) Corba (Common Object Request Broker Architecture) megvalósításai között válogathat (Visigenic VisiBroker, Iona Orbix, IBM SOM/DSOM stb.).

Nem lenne teljes a kép a Smalltalk világ megemlítése nélkül. Grafikus fejlesztői felülettel jól felszerelt, gazdag objektumkönyvtárral ellátott, tiszta (pure) objektumorientált világba csöppen a fejlesztő, ha IBM VisualAge/Smalltalk, ParcPlace Digitalk VisualWorks vagy akár más Smalltalk eszközt választ.

Az OO programozás alkalmazásának előnyei

Fokozott mértékű modularitás

A strukturált programozással kezdetét vette a moduláris felépítésű szoftverek készítésének korszaka. Az OO programozás ennek a moduláris építkezésnek a legkifinomultabb megvalósítását teszi lehetővé. Egy OO rendszerben minden folyamat objektumok együttműködéseként valósul meg. Valamennyi objektum a teljes rendszernek csak egy adott feladatát látja el, jól meghatározott interfésze van. Az objektumokra bontott rendszert könnyebb implementálni (a feladatok akár külső programozóknak is kiadhatók), egyszerűbb a tesztelés és hibakezelés, az egyes feladatokat végző objektumokat könnyen ki lehet cserélni más objektumokra, és így a bővítés, módosíthatóság „fájdalommentesebb".

Az OO technológiát valójában nem a szoftveresek, hanem a hard-

veresek (telefónia, számítógépipar stb.) találták fel: gondoljunk csak arra, hogy milyen sok gyártótól származó komponens (objektum) képes együttműködni egy számítógépen vagy telefonrendszerben! A tervezés során a következő objektumtípusokat szokás megkülönböztetni: szakterületi (domain, application), vezérlő, adatbázis-interfész, konténer (beágyazó) stb. Ez a kategorizálás segít a fejlesztőknek a rendszer objektumainak helyes megtervezésében.

Újrafelhasználás

Az elkészült, tesztelt, beüzemelt (bevált) objektumok az újabb rendszerek készítésénél kész építőkövekként (panelelemekként) újrafelhasználhatók, ezzel növelve a rendszer megbízhatóságát és csökkentve a fejlesztési költségeket.

A már említett komponensalapú fejlesztés oly mértékben előtérbe helyezi az újrafelhasználást, hogy a fejlesztési fázist két részre bontja: felhasználandó komponensek kiválasztása (beszerzése) és/vagy előállítása; a végfelhasználói alkalmazás összeszerelése.

A gyors alkalmazásfejlesztési (Rapid Application Development — RAD) módszer elképzelhetetlen lenne együttműködő újrafelhasználható objektumok nélkül. Maguknak az újrafelhasználható komponenseknek az előállítása során is alkalmazhatunk újrafelhasználható elemeket (bináris) komponensek (OCX [ActiveX], Java applet stb.) vagy osztálykönyvtárak (Microsoft Foundation Classes C++, IQSOFT Business Class Library/Centura, Metex ClassIQ/Centura stb.) beépítésével.

Objektumorientált tervezés

Az OO programozás óriási méretű elterjedésével felvetődik a kérdés: a hagyományos strukturált tervezési módszertanok megfelelnek-e az OO „filozófia" támogatására? Annyira felelnek meg, mint ahogy egy csavarhúzó nyele is alkalmas egy szög beverésére. Csak az OO elemzési és tervezési módszertanok alkalmasak igazán OO programozással előállított rendszerek tervezéséhez. Az első jelentős OO módszertanok 1988–1991 között jelentek meg.

1995-re az OO módszertanok teljesen kiforrtak, és a fejlesztési folyamat szempontjából — az egymásra gyakorolt megtermékenyítő hatás következtében — viszonylag egységessé váltak. Megszületett az egységes, általánosan elfogadott OO tervezési jelölésrendszer (Unified Modeling Language).

Valamennyi jelentős CASE (Computer Aided Software Engineering) eszközgyártó kiegészítette tervezőrendszereit az OO módszertanokat támogató részekkel (Popkin SA Object, Cayenne Object Team stb.), illetve számos gyártó csak OO tervezési módszertanokat támogató CASE eszközt készített (Platinum Paradigm Plus, Rational Rose...).

Az OO elemzési és tervezési módszertanok főleg az előállítandó szoftverrel kapcsolatos tervezést támogatják; ha valaki az üzleti folyamatot, munkafolyamatot (workflow) akarja modellezni, akkor ehhez valamilyen célszoftvert kell keresnie.

Az OO módszertan elvei szerint (pl. Rumbaugh OMT, Booch, Coad) egy OO CASE eszközzel (pl. Platinum Object Pro) létrehozott rendszer jobb minőségű, megbízhatóbb, bővíthetőbb szoftvert jelent, ha feltételezzük a fejlesztők megfelelő színvonalú hozzáértését.

Objektumorientált adatbázis-kezelés

Az objektumorientált (kliensoldali) programozáshoz jól illeszthető a szerveroldali relációs adatbázis-kezelés. Az OO tervezés és a relációs adatbázisok összhangban vannak egymással.

A Rumbaugh és társai által 1991-ben publikált Object Modeling Technique (OMT) OO elemzési és tervezési módszer sikerét jórészt annak köszönheti, hogy zökkenőmentes átmenetet teremtett az egyedtípus modellezési (entity relationship, logical data structure) technizált és az adatáramlási diagramokat ismerő szakemberek számára. Az OMT-t ismertető könyvben az egyik legjobban kifejtett rész az objektummodell relációs adatbázisbeli megfeleltetése.

Már több mint tízéves múltra tekinthetnek vissza az OO adatbázis-kezelők, de igazából csak napjainkban indulnak „hódító útjukra". A nagy lekérdezési sebességet igénylő, komplex adatszerkezetekkel dolgozó (valós idejű [real time], katonai, tervező stb.) rendszerekben régóta előnyben részesítik az OO adatbázis-kezelőket a relációsakkal szemben. Üzleti alkalmazásokhoz ez utóbbiak is megfelelnének, ám a rendszerek bonyolultságának fokozódásával ezen a területen is terjednek az OO adatbázis-kezelők, amelyek közül a legnépszerűbb az Object Design cég ObjectStore rendszere.

Látható, hogy az OO technológia a rendszertervezéstől a programozáson át az adatbázis-kezelésig az összes szoftverterületre kiterjed. Mára kiforrott; minden területen számos szállító bevált termékei állnak rendelkezésre, és Magyarországon is megjelentek azok a know-how központok, amelyek segíthetnek az OO technológiában még járatlan fejlesztőműhelyeknek a zökkenőmentes átmenetben.

Németh Miklós

Javasolt irodalom

- Booch, Grady Object-Oriented Analysis and Design with Applications. Benjamin/Cummings 1994
- Rumbaugh, James, M. Blaha, W. Premerlani, F. Eddy és W. Lorensen. Object-Oriented Modeling and Design. Prentice Hall, 1991
- Wirfs-Brock, Rebecca, B. Wilkerson és L. Wiener. Designing Object-Oriented Software. Prentice Hall, 1991
- Coad, Peter, D. North és M. Mayfield. Object Models. Strategies, Patterns and Applications. Prentice Hall, 1995
- Mowbray, Thomas J., R. Zahazi. The Essential CORBA: Systems Integration Using Distributed Objects. John Wiley & Sons, 1995
- Chappell, David. Understanding Activex and OLE. Microsoft Press, 1996

SUNaszétről — Sun-hírek a nagyvilágból

Határok nélkül

Öt szerverből és munkaállomásból álló, nagy megbízhatóságú és rendelkezésre állású számítógéprendszert szállít s integrál az ICON az MMG AM Rt. fővállalkozásában, a Prolan Kft. bevonásával futó projekt keretében. A rendszer egy ukrajnai export kőolajvezeték-szakasz teljes folyamatirányítását látja majd el.

PC helyett JavaStation — a Nomuránál

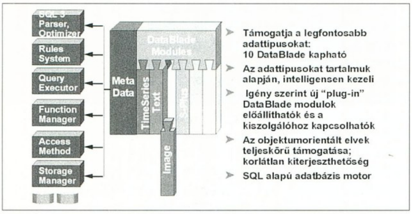
A vezető nemzetközi befektetési bank, a Nomura International mintegy 1100 JavaStation hálózati számítógépre cseréli PC-it. A Java-alapú üzleti alkalmazásokat futtató JavaStation számítógépek bevezetésére a Nomura technológiai rendszerének nagyarányú átalakításán belül kerül sor Ezzel párhuzamosan Java-alapú grafikai interfészeket is bevezetnek, amelyek révén kihasználhatóvá válnak a vállalati Java számítástechnikai modell előnyei.

A Nomura International, amely a JavaStation hálózati számítógépeket a következő két évben alkalmazza majd, jelentős költségmegtakarítást vár a számítógépek használatától a már meglévő front-end Java-alkalmazások és a back-end Solaris-alapú szerverek esetében.

„A JavaStation működtetése olcsóbb, mint a PC-ké" — nyilatkozta *Geoff Doubleday*, a Nomura International Információs Rendszerek Divíziójának ügyvezető igazgatója. „Azáltal, hogy alkalmazásainkat Javában írjuk, óriási előnyhöz jutunk a portabilitás szempontjából, hiszen az alkalmazások bármely általunk választott platformon futhatnak." A GUI-alapú alkalmazások Javában történő fejlesztésével a Nomura alkalmazásokat és szolgáltatásokat nyújthat ügyfeleinek az Interneten és magánhálózatokon keresztül, függetlenül attól, hogy azok milyen számítástechnikai platformot használnak.

Illustra: az Informix adatbázis-kiszolgálója

Az Illustra ORDBMS (objektumrelációs DBMS) a relációs adatbázis-kezelés olyan kiterjesztése, amely az alfanumerikusakon túl a gyakorlatban legfontosabb adattípusok specifikus kezelését is beépítette a kiszolgálóba. Megvalósítja az alapvető objektumorientált elveket, ezért a *kiszolgáló* (objektumorientált értelemben) *korlátlanul kiterjeszthető*: a meglévő adattípusok egymásra építhetők, a kiszolgáló újabb adattípusok kezelésével bővíthető.



Illustra: egy új DB-architektúra

Az Illustra tehát a relációs adatbázis-kezelőkkel ellentétben nemcsak az alfanumerikus adatokat érti — tárolja, indexeli, kezeli specifikumaiknak megfelelően, ad hozzájuk felhasználói műveleteket —, hanem minden olyan adattípust, amelynek a kezelésével az adott kiszolgálópéldányt felruházták. Ez a minőségi különbség teszi lehetővé, hogy a nem hagyományos adatokat az RDBMS-eknél lényegesen hatékonyabban kezelje. Az egyes adattípusokat kiszolgálóoldali osztálykönyvtárak valósítják meg, amelyeket az Illustra esetén DataBlade-eknek hívnak. Az elnevezés arra utal, hogy a kezelni kívánt adattípusok bővülésével a kiszolgálópéldányhoz olyan könnyen kapcsolható új osztálykönyvtár, mint amikor új pengét pattintunk univerzális nyélbe, foglalatba.

Az objektumorientált elvek adatbázis-kiszolgálóbeli megjelenésének okaival és a megvalósítások általános jellemzésével januári és februári számainkban (Objektumrelációs adatbázis-kezelés I–II.) már foglalkoztunk. Ezért a fenti bevezetővel megelégedve a továbbiakban az Illustra létrejöttét, tulajdonságait, a legfontosabb DataBlade modulokat és alkalmazásokat ismertetjük.

Egy kis történelem

Az Illustra ORDBMS hétévnyi kutatás és fejlesztés gyümölcse. A kutatás a Berkeley Egyetemen a Postgres* projekt keretében folyt, *Michael Stonebraker* professzor irányításával, aki 1992 augusztusában megalapította az Illustra céget, ahol az ígéretes kutatás eredményeként termékfejlesztésbe fogtak. Az Illustra ORDBMS-t 1995 végére már mintegy 400 ügyfélnél alkalmazták, és partnereik közé számított pl. az America Online, az Intel, a Silicon Graphics, a Sun és a Tandem.

** A Postgres elnevezés magyarázata az, hogy Michael Stonebraker az Ingres cég egyik alapítója, 1993-ig az igazgatótanács tagja volt.*

A Gartner Group ábránkon látható három „mágikus négyzete" 1994-től 1995 közepéig értékeli a DBMS-gyártók helyzetét. Az 1995. júniusi négyzet jól mutatja, hogy mind az Informix, mind az Illustra technológiai szempontból élen jártak, de a két cég eltérő nagyságának, anyagi erejének következtében különböző tartományokba kerültek. Az Illustra kicsi induló tőkéje miatt önmagában csak olyan lassan nyerhetett volna teret, hogy mielőtt igazán számottevővé válik, addigra a nagyok már megjelennek saját objektumrelációs termékeikkel, és így nem tudta volna meghódítani a piacot. Ezért történt meg a két technológiai vezető cég összeolvadása 1996 februárjában. Azóta az Illustra ORDBMS az Informix cég terméke, Michael Stonebraker pedig az Informix egyik elnökhelyettese és technológiai vezetője. Létrejött az Illustra új verziója, és a technológiák ötvözésével az INFORMIX-Universal Server.

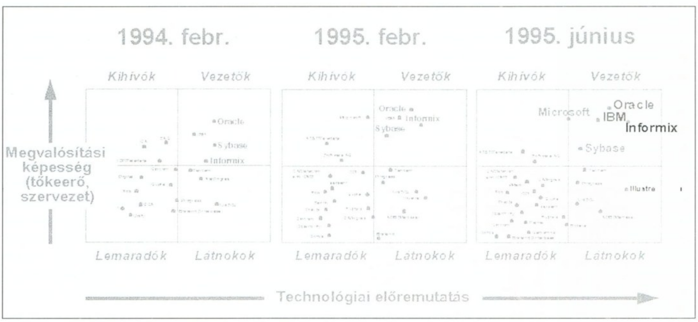
DataBlade modulok

Az Illustra tíz DataBlade modulhoz több mint száz adattípust kínál, melyek egy részét az Illustra, a többit partnercégek fejlesztették ki. A DataBlade modulok kidolgozását módszertan és fejlesztőkörnyezet teszi lehetővé. Egy-egy általánosan felhasználható területre, mint amilyen pl. a szabad szövegek, képek vagy térbeli objektumok kezelése, csak az adott témakör reprezentációs, indexelési, kezelési megoldásait kiválóan ismerő szakemberek képesek hatékonyan alkalmazható DataBlade modulokat kifejleszteni. Ezeknek az általános, horizontális termékeknek a különböző kombinációi azután nagyon sok alkalmazási területen felhasználhatók anélkül, hogy ez a DBMS működésével kapcsolatban mély ismereteket kívánna. A vertikális piacokra tehát független fejlesztőtársaságok készíthetnek alkalmazásokat.

Az Illustra DataBlade modulok a következő horizontális területekre szolgáltatnak adattípusokat:

- szabad szövegkezelés két, egymásra építhető DataBlade-del
  - keresés kifejezések alapján
  - szöveg jellemző szavainak (kulcsszavainak) a meghatározása
  - szöveghez hasonló tartalmú szövegek megkeresése
- raszterkép
  - kivágás
  - nagyítás-kicsinyítés
  - vetítés, forgatás
- álló- és mozgókép visszakeresése szín, felület, alak, összetétel hasonlósága alapján
- 2D és 3D objektumkezelés (mind térképi, mind mérnöki objektumokra)
  - létrehozás, összerakás, manipulációk
  - tartalmazás, átfedés vizsgálata
  - kerület, terület, felület, térfogat kiszámítása

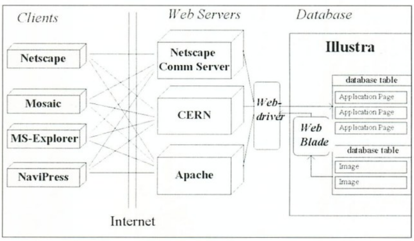
A Gartner Group DBMS-gyártókra vonatkozó értékelése



• időbeli folyamatok adatainak kezelése: idősorozatok pl. részvények árfolyamváltozásainak vagy mérési adatoknak a hatékony kezelésére
  • naptárak hozzárendelése idősorozatokhoz
  • idősorozatok feltöltése, metszetek előállítása, összehasonlítása
• statisztika: több mint 1000 függvény pl. idősorozatoknak és metszeteiknek az elemzésére
• Web kapcsolat
  • lapok dinamikus előállítása
  • tranzakciókezelés
• gateway-ek más adatbázis-kezelők (Informix, Oracle, Sybase) felé

A Web DataBlade hajtja végre a Web driver által szállított, HTML-be ágyazott SQL hívásokat, és az adatbázisból származó eredményt továbbítja a Web kiszolgálónak. A Web driver rendeli hozzá a felhasználóhoz — annak első megjelenésekor — azt a dinamikus belső azonosítót, ami az adatbázis-kapcsolatot a további párbeszéd során meghatározza, és megvalósítja az igényelt tranzakciót többlépéses párbeszéd esetén. Maga a Web driver kliens/szerver felépítésű, így lehetővé teszi az adatbázis és a Web kiszolgáló különböző gépekre helyezését, és optimalizálja a kapcsolattartást mind a Web szerver és az Illustra szerver, mind a Web kliensek és az Illustra szerver között.

Tekintsük például a 2D és a Web DataBlade együttes alkalmazását! Az Internet alkalmazás főmenüjéből eljuthatunk Európa térképéhez, ahol egérrel körbekerítve a számunkra érdekes területet, lekérdezhetjük, hogy az adott területen belül milyen szolgáltatók vannak. A szolgáltatások közül a megfelelőt kiválasztva egy — akár több Web lapot is átfogó — tranzakcióban elküldhetjük megrendelésünket. A 2D DataBlade modul a térképhez rendelt szolgáltatókat két egyenrangú dimenzióban indexeli, tehát nem az RDBMS-eknél megszokott, az összetett index komponenseit rangsoroló, linearizáló B+-fa indexelést alkalmazza.



Az Illustra Web architektúrája

Partnerek és alkalmazások

A sok száz alkalmazó és partner közül itt csak néhányat sorolunk fel, olyanokat, amelyek nem DataBlade-ek fejlesztésében működnek közre.

A Tandem licenceli az Illustra ORDBMS technológiáját.

A Netscape és az Informix kölcsönösen a másik termékével egy csomagban (Netscape LiveWire Pro, illetve INFORMIX-OnLine Workgroup Server) adja a sajátját, teljes intranet/Internet alkalmazás kifejlesztését téve lehetővé. A Netscape AppsFoundry nevű Web alkalmazás-gyűjteményének 21 tagjából 20 az Informix Workgroup Serverét használja.

A Sun szoftverpartnereinek a Weben elérhetővé tett katalógusát (*catalyst.sun.com)* az Illustra kiszolgáló, a Web és a szöveges kereső DataBlade modulok valósítják meg. Az Informix grafikus fejlesztőkörnyezetet készít Java-alapú webes adatbázis-alkalmazásokhoz.

A NASA EOS programja a Föld globális változásait kutatja. Az Informixszal 2002-ig megkötött együttműködés keretében a megfigyelő világhálózat intelligens, tartalomalapú adatbázisait az Informix ORDBMS-ek valósítják meg. A hihetetlen mennyiségű (már nem is terabájtokról beszélnek) képanyagot és mérési eredményt a Földet behálózó több ezer kutatóintézet számára kell egyidejűleg, tájékozódásra és kiértékelésre alkalmas módon hozzáférhetővé tenni.

Az Illustra tulajdonságai

Az Illustra minden objektumorientált alapelvet megvalósít. Ezek közül itt csak az öröklést emeljük ki, mert az tulajdonságok többszörös (több szülőtől való) átvételét is lehetővé teszi. A típusok belső reprezentációja és indexelése specifikus. Az adattípusokhoz felhasználói függvényeket mind SQL, mind C nyelven definiálhatunk. A költségalapú optimalizálás függvényekre is kiterjed. Ez itt sokkal fontosabb, mint az egyszerű adatokon működő relációs esetben, mert a mégoly hatékonyan megvalósított függvények végrehajtási költsége széles skálán mozog. Gondoljunk pl. a szövegek vagy képek között hasonlóság alapján kereső műveletekre.

Az Illustra a változó adatokat sajátosan kezeli. Ha másképp nem rendelkezünk, a változás időpontjával együtt (az eredeti helyen) megőrzi az objektum régi értékét is, és módot ad az objektum élettörténetének a végigjárására. Ez a lehetőség hasonlít ahhoz, mintha az értékeket egy idősorozatban tárolnánk, de általánosabb és kevésbé hatékony. Ezzel a lehetőséggel objektumok verziókezelése is megvalósítható. A feleslegesnek ítélt, elavult adatok természetesen eltávolíthatók az adatbázisból. Az Illustra minden alapvető DBMS-tulajdonságot megvalósít. Tranzakciókezelésnél az adatváltozások időpontjának megőrzése lehetővé teszi az adatbázis állapotának adott időpontig való visszagörgetését.

Az Illustra Internationalisation Release a Unicode fehasználásával módot ad nemzeti karakterkészletek specifikumainak figyelembevételére, ilyen DataBlade modulok létrehozására.

Az Illustra utóda: INFORMIX-Universal Server

Az Informix termékskáláját — sokkal inkább, mint versenytársait — az egységes architektúrán alapuló integráltság jellemzi. Az Informix relációs adatbázis-kiszolgálóinál ez a DSA (Dinamikusan Skálázható Architektúra), amely munkaállomásoktól munkacsoportos kiszolgálókon és nagy SMP gépeken keresztül a lazán kapcsolt és masszívan párhuzamos gépekig minden RDBMS igényt hatékonyan kielégít, beleértve az OLTP, az OLAP és az adatáruházi intranet/Internet alkalmazásokat. A hálózatba kötött kiszolgálók

• az osztott adatbázis-kezelés és a replikálás gazdag választékát kínálják heterogén (nem tisztán Informix) környezetben is,

• egy helyről, közös felületről adminisztrálhatók.

Az Illustra természetesen nem DSA architektúrájú; az Informix relációs kiszolgálóihoz csak gateway-en keresztül kapcsolható. Az Illustra objektumrelációs és az Informix DSA technológiája azonban kölcsönösen kiegészíti egymást, és jól ötvözhető. Ezt felismerve olvasztotta magába az Informix az Illustra céget 1996 februárjában, s az Informix korábbi objektumrelációs fejlesztéseit felhasználva rögtön hozzáláttak a két technológia általánosításához és egyesítéséhez. 1996. december 3-ára pedig elkészült az INFORMIX-Universal Server.

Az új kiszolgálóval az Informix termékskálája újra egységessé vált. Ez az első, az alapvető objektumorientált elveket mind megvalósító, korlátlanul kiterjeszthető adatbázis-kiszolgáló, amelyet vezető DBMS-gyártó készített.

Az Informix Weben elérhető információit *(www.informix.com)* az Universal Server szolgálja ki a hozzá igazított Web DataBlade segítségével. Az adatbázisban tárolt szövegek közül tartalmuk alapján sorolthatjuk föl a kereső kifejezésünkhöz illeszkedőket; a szöveghivatkozások dinamikusan generálódnak. A szabad szöveges keresést a Verity cég Topic DataBlade-je valósítja meg.

Az INFORMIX-Universal Serverhez eddig 29 DataBlade készült el. Az Universal Serverről, valamint az elérhető DataBlade-ekről további cikkekben számolunk be.

BALOGH KÁLMÁN
KBALOGH@INFORMIX.HU

Hálózatkezelés Jávában

A Jáva hálózati könyvtára (java.net) jelenleg a TCP/IP protokollcsaládra épül — ez a szorosan vett Internet hálózat alapja is —, bár ennek részleteit csaknem teljesen elfedi a programozó elől, így elvileg nem kizárt, hogy más alapprotokollokat használó könyvtár-implementációk is megjelenjenek.

Cikkünkben a korlátozott terjedelem miatt a szállítási szintű protokollok közül csupán a TCP kapcsolatorientált protokoll használatával foglalkozhatunk, de érdemes megemlíteni, hogy a *java.net* könyvtár a datagramokra épülő UDP protokollt is támogatja. Ezenkívül e pillanatban az alkalmazói szintű protokollok közül csak a Web alapját jelentő HTTP *(Hypertext Transfer Protocol)* protokoll közvetlen használatát támogatja, viszont lehetőséget ad a programozóknak, hogy egyéb kezelőeljárásokat *(protocol handler)* is beilleszthessenek a rendszerbe. A programkákhoz tartozó néhány speciális hálózattal kapcsolatos tevékenység is megtalálható vagy a könyvtárakban vagy magában a virtuális gépben.

Szállítási szintű összeköttetés kezelése

A TCP/IP hálózat szállítási rétege kétfajta, ún. összeköttetés-alapú és összeköttetés-mentes szolgáltatást ismer. A összeköttetés-alapú szolgáltatást megvalósító TCP *(Transfer Control Protocol)* protokoll a kommunikáló programok között tulajdonképpen egy kétirányú csővezetéket hoz létre, amelyen az információ *Byte*-ok formájában áramlik. A kapcsolatorientált szolgáltatást megbízhatónak is nevezik, mert a TCP biztosítja a Byte-folyam torzításmentes, sorrendhelyes átvitelét, kiküszöbölve az elveszett, megsérült, megduplázódott, késve érkezett információcsomagok hatását. Ha pedig semmi nem segít, akkor legalább a küldő, de néha a vevő is értesül arról, hogy valami helyrehozhatatlan hiba történt.

Az alkalmazások egy részének nincs szüksége ekkora megbízhatóságra, ezek beérik az UDP *(User Datagram Protocol)* protokoll összeköttetés-mentes szolgáltatásával. Ebben az esetben a könyvtárak és a hálózati architektúra mindössze annyit ígér, hogy az egy — nem túl nagy — csomagba összepakolt Byte-okat egyben elküldi a címzettnek, és mindent megtesz azért, hogy az oda is találjon; viszont nemigen törődik az előforduló hibák kiküszöbölésével.

Mivel mindkét esetben konkrét programok kommunikálnak, ahhoz, hogy ezek egymásra találjanak, a számítógéphez tartozó egyedi hálózati címet — az ún. IP címet — ki kell egészítenünk a gépeken futó programok megkülönböztetésére szolgáló címmel. A TCP/IP hálózati rendszerben nem a programokat, hanem a kommunikációra szolgáló kapcsolódási pontokat, az ún. kapukat *(port)* azonosítjuk. Általában korlátozzák az első 1024 kapu címhasználatát, csak rendszergazda-jogosultságú programok olvashatják, a többi a „mezei" felhasználók rendelkezésére áll. Az ún. jól ismert kapuk *(well-known ports)* minden számítógépen szabványos funkciókat betöltő programokhoz tartoznak.

Egy tipikus szerver működésének lépései:

1. Az egyes kapukhoz tartozó kommunikációs csatornákat *socket*eknek nevezzük. Kapcsolatorientált kommunikációnál kétfajta socket használatos: a szerveroldalon a kiépülésre váró kapcsolatot az ún. *ServerSocket* osztály egy példánya valósítja meg:

```
ServerSocket server = new ServerSocket(port);
```

2. A kliens/szerver rendszerekben a szerver passzív szerepet játszik, arra várakozik, hogy egy kliens megszólítsa. Ha sikeresen kiépül a kapcsolat, a szerver visszatér az *accept* hívásból a kiépült kapcsolatot reprezentáló Socket példányból:

```
Socket socket = server.accept();
```

3. A socketből két adatfolyam, az *input* — a kliensből a szerver felé irányul — és az *output* — a szerverből a kliens felé tart — kapható meg:

```
InputStream input = socket.getInputStream();
OutputStream input = socket.getOutputStream();
```

4. A szerver törzse az így megszerzett input adatfolyamot olvassa, a vett információt értelmezi, és esetleg az output folyamon válaszol.

5. Az adatcsere végén a szervernek vagy a kliensnek le kell zárnia a kapcsolatot jelentő socketet. Ehhez először az adatfolyamokat illik lezárni.

Ezután minden indulhat elölről, pontosabban a 2. lépéstől, ahol a szerver új kérésre vár. Amennyiben egyidejűleg több kérés kiszolgálására is szeretnénk felkészíteni szerverünket, akkor a 3–5. lépéseket egy-egy külön szállal *(Thread)* érdemes végrehajtatni, miközben az eredeti szál a 2-es lépésben várakozik. A kliensoldali program még ennél is egyszerűbb. Először felvesszük a kapcsolatot az adott Internet címmel (vagy névvel) ellátott számítógép megfelelő kapuján várakozó kiszolgálóprogrammal:

```
Socket socket = new Socket (host, port);
```

Innen a lépések megegyeznek a szerveroldal 3–5. lépéseivel.

Általános szerver és felhasználása

Példánk egy általános szerverprogramot mutat be. Általános abban az értelemben, hogy a szerver tevékenységét egy absztrakt osztály végzi, a konkrét szervert megvalósító, leszármazott osztályban csak egy konstruktort és egy módszert kell megírnunk. Szerverünk párhuzamos működésű, azaz a hozzá tartozó kapun folyamatosan várja a klienseket, egy kiépülő kapcsolat kiszolgálására saját szálat indít el. Így egyidejűleg több klienssel is kommunikálhat.

(Megjegyzés: terjedelmi korlátok miatt a programot kissé megkurtítottam. A hálózati hibák részletesebb kezelése és a szerver kulturált leállítása szenvedte a legtöbb csorbát.)

TCPListener osztály

Feladata a kapcsolat figyelése és a párhuzamos kiszolgálószálak létrehozása:

```
import java.net.*;
import java.io.*;
import java.util.*;
```

Az osztály minden egyes példánya megvalósít egy TCP szervert. Ehhez párhuzamos szálat indítunk el, hogy ne kelljen a konstruktorban várakozni mindaddig, amíg a szerver le nem áll:

```
public class TCPListener implements Runnable
{
 private ServerSocket server;
 private int port;
 private TCPServerFactory factory;
 private Thread listener;
 private Vector servers;
```

A konstruktor megkapja annak a kapunak a címét, ahol a szerver a kapcsolatokat várja. A második paraméter egy speciális objektum, amely képes arra, hogy *TCPServer* típusú objektumokat állítson elő (ezért nevezik gyárnak, *factory*nak). Ezek az objektumok a figyelőszállal párhuzamosan bonyolítják majd le az egyes kliensekkel a kapcsolatot. A konstruktor a paraméterek egyedváltozókba tárolása után elindítja a kapcsolatra figyelő szálat:

```
public TCPListener(int port, TCPServerFactory
factory)
{
 this.port = port;
 this.factory = factory;
 listener = new Thread(this);
 listener.start();
}
public void run ()
{ try {
```

A hálózati program első lépései az általános modellt követik. A szerverkapu létrehozása után a kiszolgáló végtelen ciklusban accepttel egy kliens által kezdeményezett kapcsolatra vár. Ha ez megjött, a kiszolgálógyárral elkészíttetünk egy újabb kiszolgálóobjektumot:

```
server = new ServerSocket(port);
while (true) {
 Socket socket = server.accept();
 TCPServer server = factory.createServer(socket);
}
```

A hálózati kapcsolat kialakítás közben előforduló összes hibát együttesen kezeljük — ebben az esetben elhanyagoljuk —, a kiszolgáló leáll:

```
 } catch (IOException e) {}
 shutdown();
}
```

A *shutdown* módszer feladata a figyelőszál és az élő kiszolgálók teljes leállítása:

```
 public void shutdown()
 {
 if (listener != null) listener.stop();
 }
}
```

Absztrakt TCPServer osztály

Feladata az egyes kliensektől érkező kapcsolatok kezeléséhez szükséges adminisztráció lebonyolítása:

```
import java.net.*;
import java.io.*;

abstract public class TCPServer implements Runnable
{
 private Socket socket = null;
 private Thread life = null;
 protected DataInputStream in = null;
 protected PrintStream out = null;
 protected TCPServer (Socket s) throws IOException
 {
  socket = s;
```

Az osztály konstruktora egyedváltozókban tárolja a paramétereit, majd megszerzi a kiépült kapcsolatot jellemző ki- és bemeneti adatfolyamot. Érdemes megfigyelni, hogy a *get...Stream* eljárásokból visszakapott egyszerű adatfolyamokat — kényelmesebb kezelésük érdekében — különböző szűrőkön vezetjük át. A *Data...* és *Print...* osztályok a beépített Jáva adattípusok közvetlen be/kivitelét valósítják meg:

```
in = new DataInputStream(socket.getInputStream()));
out = new
PrintStream(socket.getOutputStream()),true);
```

Az új szálból elindulása előtt démont csinálunk, azaz olyan szálat, amelyet a szerver leállásánál a Jáva virtuális gép magától megszakít:

```
 life = new Thread(this);
 life.setDaemon(true);
 life.start();
}
```

A kiszolgáló tevékenysége a *run* eljárásban zajlik. Ez nem más, mint egy általános *handleSession* módszer meghívása. Az esetleges hibákat a program elkapja, de nem kezeli, hiszen nem is tudna velük mit kezdeni:

```
public void run()
{ try {
   handleSession();
  } catch(IOException e) {};
  shutdown();
}
```

A szerver lezárása egyszerű — mind a nyitott adatfolyamokat, mind a socketet lezárjuk:

```
public void shutdown()
 { try {
  if (in != null) in.close();
  if (out != null) out.close();
  if (socket != null) socket.close();
 } catch (IOException e) { }
}
```

Szeretnénk általánosan megírni szerverünket, amely lehetőleg az egyes kiszolgálószálak adminisztrációjának minden részletét kezeli, a programozónak csak a kapcsolat konkrét lebonyolításával kell foglalkoznia. Erre szolgál az itt következő handleSession absztrakt módszer. Ezért TCPServer osztályunknak is absztraktnak kell lennie, de az ebből leszármaztatott osztályok majd definiálhatják a szükséges módszer törzsét.

A módszer belsejében használhatjuk a nyitott *in* és *out* adatfolyamokat. A módszer lenyomatának tanúsága szerint a belsejében az átviteli hibákkal nem kell foglalkoznunk, ezt a felső szint (az előbbi run módszer) elvégzi:

```
abstract public void handleSession() throws
IOException;
}
```

Kiszolgálógyár

Ez a legérdekesebb ötlet az egész programban. Az elv ismerős, az objektumorientált programozásban újabban nagyon divatos módszertan, az ún. tervezési minták *(design pattern)* egyik ismert tagja a „gyár". Akkor használjuk, amikor azonos objektumokat kell készíteni, ám a program fordításakor még nem ismerjük ezek pontos típusát. A megoldás: definiálunk egy általános „gyár" osztályt, amely adott — esetünkben a *createServer* eljárása —, segítségével általános objektumpéldányokat — itt TCPServer típusúakat — gyárthatunk. A program futásánál aztán majd az általános gyár helyett egy konkrétat használunk, amelyik az általános eredménytípusnak megfelelő, abból leszármaztatott típusú objektumokat állít elő. Jávában az ilyen általános osztályokat interfészek segítségével is megadhatjuk. Esetünkben ez különösen kézenfekvő megoldás, mert a módszer belsejét úgysem tudnánk itt megírni:

```
import java.net.*;
```

(folytatás a 21. oldalon)

Java WorkShop

A Jáva programozási nyelv rohamos terjedésével együtt egyre inkább megfogalmazódott az igény, hogy a programozók a JDK által nyújtott fejlesztési lehetőségeket meghaladó fejlesztői környezetet használhassanak programjaik létrehozására, tesztelésére.

Egyre-másra jelentek, jelennek meg az ún. integrált fejlesztői környezetek *(Integrated Development Environment, IDE)*, amelyek a Jáva programok létrehozásának minél több lépését próbálják lefedni. Néhány ilyen környezet (a teljesség igénye nélkül, hiszen gomba módra szaporodnak): *Symantec Café, Microsoft Visual J++, Metrowerks CodeWarrior* és végezetül cikkünk tárgya, a *Sun Microsystems Java Workshop* környezete.

Legfontosabb tulajdonságai

Nehéz az ilyen komplex programrendszereket összehasonlítani, már csak azért is, mert a fejlesztőeszközök megítélésénél az egyes programozókat nagyban befolyásolják korábban megszokott eszközeik, programozási szokásaik. Én sem szeretnék hát összehasonlító értékelést adni, inkább csak a JavaWorkshop tulajdonságait taglalom, az értékelést, véleményformálást inkább a programok felhasználóira hagyom.

Jávaban implementált

Mindenesetre a JavaWorkshopot a többitől elkülönítő tulajdonsága, hogy a programot teljes egészében Jávában implementálták. Ez persze nagy előnyt jelent a környezet hordozásánál, a Sun jelenleg is három platformon forgalmazza a minden platformon teljesen azonos képességű, megjelenésű programot: *Solaris 2.x, Windows NT, Windows 95*. Persze egyéb platformra is áttehető lenne, de a cég ezzel talán még a felhasználói igényekre vár. Sajnos a Jáva nyelvű implementáció hátránya a viszonylagos lassúság, a Sunnak egyelőre nincs egyetlen felsorolt architektúrán sem elegendően stabil röptében fordítója, hogy a program futását felgyorsítsa. De ami késik, nem múlik, és a program megvásárolásával — 100 dollár körüli áron — egy esztendőn keresztül az újabb változatokat is megkapjuk.

Web böngésző kezelői felület

A fejlesztői környezet érdekessége, hogy kezelői felülete egy Web böngészőre hasonlít. Nem véletlenül, hiszen a környezet valóban a *HotJava* böngészőre épül, a program kezelése általában HTML lapokon keresztüli navigálással történik, az egyes funkciókat a lapokba ágyazott Jáva programkák hajtják végre.

Aki már böngészett a Weben, annak ismerős lesz a felület, bár ez nem feltétlenül azt jelenti, hogy könnyen használható. Ilyen méretű programnál még a leginkább felhasználóbarát felület sem pótolhatja a funkciók, lehetőségek alapos megismerését.

Egyébként a környezet mint Web böngésző is használható — bár képességeiben elmarad az elterjedt böngészők új verzióitól —, például az összes online dokumentáció is HTML formájú állományokból olvasható.

A platformfüggetlenségnek az az ára, hogy a program vizuális megjelenése nem hordoz pl. Windows 95 sajátosságokat, így nem feltétlenül olyan „szép", cizellált, mint a csak egy platformra megírt programnál esetleg elvárható.

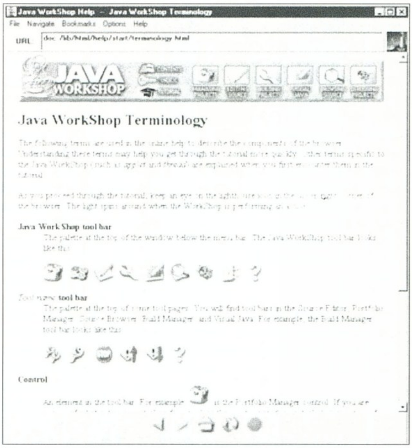
Főbb komponensei

A fejlesztői környezet attól integrált, hogy a programfejlesztés több lépésének támogatását egyetlen eszközbe ötvözi, az egyes funkciók közötti összhangot, váltásokat többé-kevésbé harmonikusan megvalósítva. A legfontosabb programösszetevők:

- *Project Manager* — a Jáva programhoz szükséges összes állomány rendezését, együttes kezelését biztosítja;
- *Build Manager* — a megváltozott állományok újrafordítását végzi;
- *Project Tester* — automatikusan, a fejlesztői környezet elhagyása nélkül futtatja elkészült programunkat vagy programkánkat;
- *Portfolio Manager* — segítségével projektjeinket és azok eredményeit mappákba *(portfolio)* foghatjuk össze, és az Internet/intraneten keresztül hozzáférhetővé tehetjük egy fejlesztői csoport vagy a felhasználók számára;
- *Debugger* — a programok futásának követését támogatja;
- *Source Editor* — a forráskód írásához, módosításához, a programváltozatok karbantartásához;
- *Source Browser* — a programunk és a Jáva könyvtárak struktúrájának böngészéséhez;
- *Visual Java* — segítségével interaktívan tervezhetjük meg programunk kezelői felületét;
- *On-line Help* — hipertext dokumentációkat és a program használatát ismertető, tanító bevezetőket is tartalmaz.

Néhány érdekesség

Terjedelmi okok miatt a teljes környezetet nem lehet részletesen ismertetni, inkább étvágygerjesztőül szeretném kiemelni néhány izgalmasabb tulajdonságát, képességét.



Forráskódkezelés

A forráskódszerkesztő az egész programrendszer talán legkevésbé attraktív, korszerű része, megjelenésében, képességeiben alig több, mint egy egyszerű szövegszerkesztő. Viszont érdekessége, hogy képes együttműködni az adott platformon használatos, elterjedt forrásprogram-könyvtár rendszerekkel *(version control)*, mint pl. a *Win95/NT*-n a *PVCS*, a *Solaris*on az *RCS* és *SCCS* programokkal, lehetővé téve nemcsak a különböző programverziók karbantartását, de egy projekten több fejlesztő egyidejű munkáját is.

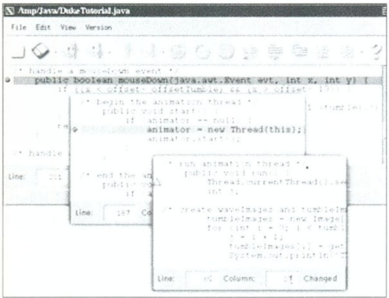
A szerkesztővel jól integrált a forrásprogram-böngésző, amelynek segítségével a programunk osztályhierarchiáját járhatjuk be, megtekintve az egyes osztályok módszereit, változóit.

Fordítás

A *Build Manager* nyomon követi, hogy a legutóbbi fordítás óta a projekt mely állományai változtak meg, és kívánságra csak ezeket fordítja újra. A fordítás során előállt hibákon sorban végiglépkedhetünk, és a forráskódszerkesztőben rögtön javíthatunk is. Sikeres fordítás után azon nyomban indíthatjuk is innen a programot.

Nyomkövetés

A lefordított programot szabadon futtatása helyett a nyomkövető fel-

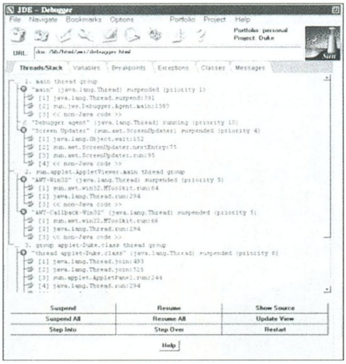


ügyelete alatt is vizsgálhatjuk. Ekkor a szokásos forráskódszintű nyomkövetők szolgáltatásait használhatjuk: töréspontok, lépésenkénti futtatás — eljáráshívásokba belépve, ill. átlépve —, változók és veremkép vizsgálata, változók értékének módosítása. A nyomkövető érdekessége, hogy mivel a Jáva képes párhuzamos szálak futtatására, a nyomkövető is felkészült erre: a töréspontoknál valamennyi szál állapotát különkülön is vizsgálhatjuk, illetve egyes szálak futását ideiglenesen felfüggeszthetjük, majd újraindíthatjuk

Kezelői felület készítése

A *Visual Java* programkomponens segítségével interaktívan tervezhetjük az alkalmazások kezelői felületét *(GUI)*. Lehetőségünk van ún. keretek *(frames)* — alkalmazás fő ablakának —, panelok — általános *GUI* elemek gyűjteményének —, illetve párbeszéd-ablakok tervezésére. Az eszközoszlopon a JDK-ban definiált összes GUI komponens, illetve konténer a rendelkezésünkre áll, ezekből az egér segítségével építkezhetünk, vagy általános tulajdonságaikat külön-külön is megváltoztathatjuk. Az összetevők elhelyezésénél a program nem abszolút koordinátákhoz pozicionál — ami nagyban megnehezítené a különböző architektúrák, különböző felbontású megjelenítők használatát —, hanem kihasználja a grafikus könyvtárban definiált tördelők *(layout manager)* szolgáltatásait.

Az elkészült grafikus felületet, amelynek leírása egy *.gui* állományba kerül, fordítás nélkül is tesztelni lehet, a *Build Manager* viszont fordításnál ebből állítja elő a szükséges Jáva forráskódot.



Projektek és mappák

A programozók egy-egy feladat megoldásához a Java WorkShopban projekteket hozhatnak létre. A projekt tartalmazza az összes megoldáshoz tartozó állományt, legyenek azok Jáva forrás- vagy lefordított állományok, GUI leírás, HTML állományok, illetve a bennük használt GIF képek.

Természetesen a projektekhez tartoznak egyéb kiegészítő információk is, mint pl. a fordításhoz használt kapcsolók, a *CLASSPATH* környezeti változó értéke, a felhasznált könyvtárak neve stb. Az egyes projektek létrehozásuk után együtt is kezelhetők, pl. másolhatók, menthetők, fordíthatók és futtathatók.

A projekteket mappákba foghatjuk össze, ezeket aztán „publikálhatjuk", azaz a helyi gépen, lokális hálózaton vagy az Interneten mások által is hozzáférhetővé tehetjük. Más programozók az általuk elérhető mappákból kivehetnek és használhatnak különböző projekteket.

K. I.

(folytatás a 19. oldalról)

```
interface TCPServerFactory
{
 TCPServer createServer(Socket socket);
}
```

Visszhang

Általános szerverünk felhasználására egy olyan példát mutatok, amely soronként visszaküld mindent a kliensnek. A szerver leáll, ha *„exit"* üzenetet vesz:

```
import java.io.*;
import java.net.*;
```

Szerverünket az általános TCPServer osztályból származtatjuk le. Csak egy konstruktort, illetve az absztrakt handleSession módszert kell megírnunk. Hasonló könnyedséggel írhatunk bonyolultabb kiszolgálókat is:

```
class EchoServer extends TCPServer
{
```

A konstruktor egyszerűen lehívja a szülő konstruktorát:

```
 EchoServer (Socket socket) throws IOException
 {
 super(socket);
 }
```

Itt kell megadnunk a szerverszálak tényleges tevékenységét. Mivel a módszer belsejében már hozzáférhetünk a két megnyitott adatfolyamhoz, csak annyi a dolgunk, hogy az egyikről olvasunk egy sort, majd kiírjuk a másikra, mindaddig, amíg — kis- és nagybetűket tetszőlegesen tartalmazó — „EXIT" szöveg nem érkezik. Egyedül arra érdemes figyelni, hogy a kapcsolat elvesztésével előfordul, hogy szöveg helyett *null*t olvasunk:

```
 public void handleSession() throws IOException
 { String s;
 do {
 s = in.readLine();
  if (s != null) out.println(s);
  } while (!(s == null ||
s.toUpperCase().equals(„EXIT")));
}
```

Szerverünk kipróbálásához elindítunk egy többszálú szervercsonkot az 5555-ös kapun, megadva neki egy olyan „gyárat", amely kiszolgáló-objektumainkat termeli majd:

```
 public static void main (String[] args)
 {
  new TCPListener(5555, new EchoServerFactory());
 }
}
```

A gyár definiálása hasonlóan egyszerű, hiszen csak egyetlen módszert kell megadnunk. A termelőeljárás lehívja a szerverosztály konstruktorát. Ha ez esetleg hibát okoz, akkor a termelő null-lal tér majd vissza:

```
class EchoServerFactory implements TCPServerFactory
{
 public TCPServer createServer(Socket socket)
 { try {
  return new EchoServer(socket);
  } catch (IOException e) { return null; }
 }
}
```

A szerver kipróbálásához használható programmal adós maradok, de azt hiszem, szorgalmi házi feladatként bárki könnyen megírhatja. Segítségképpen lapozzanak a cikk elejére, ott találhatók egy kliens működésének általános lépései.

KISS ISTVÁN

Először Magyarországon!

IBM hálózati szolgáltató központ

Európában Magyarországon vezeti be először új, globális és a maga nemében egyedülálló szolgáltatását az IBM. Ennek lényege az IBM Business Computing Utility (BCU — üzleti számítástechnikai nyilvános szolgáltatás) hálózati számítástechnikai központ létesítése és működtetése, erőforrásainak hálózaton történő igénybevétele jellemzően kis- és közepes hazai vállalkozások részére, amelyek saját erejükből nem képesek az állandóan változó, növekvő külső igényekhez saját gépparkkal és azon futtatott alkalmazásokkal igazodni. A BCU, amely az IBM Global Services nemzetközi hálózati számítástechnikai szolgáltatásainak valamennyi előnyét ki fogja használni, *Báti Ferenc* vezetésével ez év második negyedévére lesz elérhető. A BCU igazgatója az új távlatokat nyitó IBM-kezdeményezés egyes kérdéseiről nyilatkozott lapunknak.



Ettől a számunktól kezdve egy új, *TV3 rovatot* indítunk útjára, amelyben a TV3 *Negyedóra* című műsorának keddi és szerdai, informatikával foglalkozó riportjai közül dolgozunk fel párat nyomtatott formában is. Íme az első! Hasonlóképpen az Infopen hálózatokról, kormányzati és vállalati informatikáról szóló cikkeinek sorából néhány a Negyedóra adásaiban is terítékre kerül majd.

Az IBM SP2 számítógépre, a Matáv menedzselt bérelt vonalaira és kezdetben a Mikro-Volán Elektronika saját fejlesztésű Libra integrált pénzügyi, számviteli, ügyviteli programcsomagjára (majd később más, pl. Lotus Notes, EDI stb. alkalmazásokra is) alapozott „hálózati szolgáltatási iroda" jelenleg húsz, az év végére pedig várhatóan ötven fővel áll az ügyfelek rendelkezésére. Báti Ferenc szerint technológiai szempontból semmi akadálya nincs ilyen szolgáltatás nyújtásának központi erőforrásokból. Az alapkérdés: az üzleti/ügyviteli menedzsmentrendszerek fejlesztési vagy az ügyviteli folyamatok reengineering vonatkozásainak az idő- és pénzráfordítási igényei; továbbá az EU-kompatíbilis metodikák kialakítása a vállalati folyamatokban, az olyan hazai kis- és közepes cégeknek sokszor nem EU-tag versenytársakkal folytatott üzleti versenye, amelyeknek nincs több tucat emberük és 8-12 hónapjuk egy korszerű rendszer bevezetésére és üzemeltetésére. Ezekre a kihívásokra kínál megoldást az IBM BCU szolgáltatása, amely a korábbi évtizedekből ismert bérmunka-számítóközpontoktól nem csak azok online természetében tér el. Az IBM által folyamatosan bevezetett alkalmazási rendszereket úgy telepítik majd, hogy a közöttük való átjárhatóság és a rendszerek integráltsága megadja az ügyfeleknek a természetes növekedés lehetőségét. Ugyanakkor olyan alkalmazásokból szeretnének szolgáltatni, amelyek vélhetően sok felhasználónak nyújtanak megoldást. Az EU-követelményeket, funkciókat pedig — együttműködve a hazai hivatalos gazdasági, államigazgatási stb. szervezetekkel, hatóságokkal — az IBM mindig maga építi be a rendszerekbe, a felhasználó tehát automatikusan jut EU-konform alkalmazásokhoz, készíthet ilyen alapon megoldásokat.

A BCU működése abban tér el a régi számítóközpontokétól, hogy az idő előrehaladtával az IBM szolgáltatásában egyre több széles körben népszerű, hasznos alkalmazás áll az ügyfelek rendelkezésére. Báti Ferenc szerint az outsourcing és az IBM BCU szolgáltatása közötti különbség abban áll, hogy míg az előbbinél a teljes feladatot (esetleg az ügyfél erre kijelölt embereit is) átveszi a megoldást végző informatikai cég tetemes összegű szerződésért, öt-tíz vagy még ennél is több évre, addig a BCU-val a felhasználó alacsony belépési költség mellett használhatja az IBM hálózati erőforrásait. A BCU bizonyos feladatokat átvállal a megbízótól, aki azonban bármikor megváltoztathatja, felmondhatja az IBM-mel kötött szerződését. Ugyanakkor a BCU esetében — az outsourcinggal ellentétben — mindig az aktuális felhasználás mértékében tud gazdálkodni az igénybe vevő az erre a célra fordítandó pénzeszközeivel, költségekkel.

Az IBM azért választotta Budapestet a BCU bevezetésének helyszínéül, mert dinamikusan fejlődő, vállalkozó ország vagyunk, megfelelő a hálózati infrastruktúra, a helyi jogszabályok kedvező feltételeket kínálnak a szolgáltatás beindításához. Régiónk országaiban az EU-konform informatikai megoldások iránt is nagyobb az igény, amire ugyancsak szerfölött alkalmas a BCU. „Az IBM döntésében az is szerepet játszott, hogy a gazdasági és piaci mechanizmusok fejlettségét tekintve Magyarország vezető helyet foglal el a térségben. Itt kisebb befektetéssel lehet egy ilyen központot üzemeltetni, mint Nyugaton, de hozzá kell tennem, hogy azért három-négy év alatt kb. ötmilliárd forintos beruházásra mindenképpen szükség van a sikeres működéshez" — mondta Báti Ferenc.

A BCU technologizálása egyedi, hiszen még nincs rá precedens; rendkívül fragmentált piaci bevezetési folyamatok technologizálásáról van ugyanis szó. Az egy központból kínált sokféle üzleti menedzsmentrendszer egységes technologizálása számottevő költségekkel jár az IBM részére, ugyanakkor az e téren Magyarországon elért eredményeket azonnal hasznosíthatják a leányvállalatok. Ezért is lényeges, hogy az IBM globális cégként meg tudja osztani az ilyen irányú költségeket, ami nagyon fontos előfeltétele az új szolgáltatás sikerének. Jól példázza a hazai érdeklődést, hogy a februári bejelentés idején már több IBM-ügyfél tesztelte a szolgáltatást, és újabb szoftveres partnercégek kezdtek tárgyalásokat az IBM BCU-val arról, hogy alkalmazási keretrendszereiket a központon keresztül is „értékesítsék".

Kovács Attila



Felejtsen el bennünket!

- Önnek minden bizonnyal megbízható, távoli menedzsmenttel rendelkező, optimálisan üzemeltetett szerverei vannak. Ezeknek az alkalmazás-, adatbázis-, CD-, fax-, file-, kommunikációs-, nyomtató- és WEB szervereknek a védelme, karbantartása és szoftverkövetése megoldott.
- Az Ön beruházásai a körültekintő tervezés és gondos kivitelezés eredményeként értékállóak. Nincs szüksége helyi hálózatának bővítésére. Erőforrásai nagy sebességű kapcsolaton (Ethernet switch, Fast ethernet, ATM) keresztül érhetők el. Biztonságos, jól ellenőrizhető kapcsolata van a külvilággal.
- Kiválóan képzett felhasználók és a kiforrott üzemeltetési előírások együttese biztosítja a nyugodt, magabiztos munkavégzést.

Ha mégis kételyei vannak, forduljon hozzánk!

1148 Budapest, Fogarasi út 10-14. , Telefon: *252-3444, 467-0117 , Fax: 363-3659

Budapesti randevú Anders Hedbringgel, az IBS egyik alelnökével

ASW a legjobbak között

A svéd IBS cég három éve van jelen a magyar piacon az IBM AS/400 gépekre kifejlesztett ASW (Application Software) nevű integrált vállalatirányítási programrendszerével, amelyet eddig öt helyen helyeztek üzembe: a műanyag alkotórészek gyártásában érdekelt Rehau Kft.-nél, a Hitelgarancia Rt.-nél, a Győrtej és Baranyatej vállalatoknál; Salgótarjánban pedig az IBS Hungary az IBM Magyarországgal közösen ingyen bocsátotta a főiskola rendelkezésére a programcsomagot, oktatási célra. Néhány hete Budapesten járt *Anders Hedbring*, az IBS nemzetközi eladásokért felelős alelnöke, aki rövid interjút adott lapunknak.



Az IBS rendszerét a hasonló európai programcsomagok között az egyik legjobbként tartják számon. Hová sorolná az ASW-t más konkurens termékekkel szemben?

A. H.: Teljes megoldást kínálunk a vállalati általános adminisztráció terén. Az ASW előnye, hogy már megalkotásakor figyelembe vettük a nemzetközi igényeket, és azóta is ennek megfelelően kezeljük. Az elérhető tíz nyelv valamelyikén ma több mint kétezer helyen fut az IBS rendszere, amely kis, közepes és nagy cégek integrált vállalatirányítására egyformán alkalmas. Noha elsősorban iparvállalatoknak ajánljuk, a magyarországi referenciák sokfélesége is bizonyítja, hogy a termék testhezálló megoldást nyújt a termelőszféra különböző területein és az oktatásban egyaránt. A rendszer nagyszámú pénznemet kezel, más, hasonló célú szoftverekkel szemben sok nyelven megtalálható lokalizált változata. Szerveroldalon az AS/400-at és az OS/400-at, kliensoldalon pedig különféle gépeket és operációs rendszereket képes támogatni. Négy fő vállalati alkalmazási területet fog át: pénzügy, disztribúció, gyártás és adatáruház. Átlagosan 18 hónaponként új rendszerváltozatokat dobunk piacra, de időközben különböző új kliensoldali alkalmazásokkal is megjelenünk. A mintegy tíz legfontosabb versenytársunk között implementáció-sebességben egyértelműen a legjobbak vagyunk. Ugyanakkor pénzügyi rendszerünk is rendkívül versenyképes, éppen a beleépített nemzetközi pénzkezelési funkciók miatt.

Tudomásunk szerint az IBM 10%-os tőkerészesedést mondhat magáénak az IBS-ben, és a két cég szorosan együttműködik a következő generációs integrált vállalatirányítási rendszer kifejlesztésében.

A. H.: Valóban, az IBM-mel és másokkal is részt veszünk az ezredforduló utáni meghatározó rendszerek kidolgozásában, az ún. San Francisco Project (vagy másképp: IBM Shareable Frameworks) néven ismert munkacsoportban.

Ennek tevékenysége arra irányul, hogy platformfüggetlen eszközökkel és módszerekkel elkészítse azokat a szoftver-keretrendszereket, amelyeket a vállalatirányítási rendszerek fejlesztésében és azok testre szabásában lehet használni. A terv megszületését több tényező indokolta. Az egyik a vevők igénye a gyors alkalmazkodásra képes, azaz nagyon hamar testre szabható rendszerekre. A forgalmazócégek világszerte szembe találták magukat a költségek csökkentésének, a rövidebb piacra jutási időnek, a nemzetközi műveletek, a hatékonyság növelésének és a vevők jobb kiszolgálásának követelményével. Rákényszerültek, hogy radikálisan átgondolják, mit is csinálnak, és hogyan, mennyi pénzből oldják meg feladataikat. Ehhez nemcsak gondolkodásukat, hanem szervezeteiket és azok működését is át kell alakítani. A jelenlegi rendszerek nagy részét a nyolcvanas években vagy még előbb fejlesztették ki, az akkori színvonalnak megfelelő eszközökkel. Ez alól az IBS sem kivétel. RPG programnyelven és az utóbbi időben CASE eszközökkel készítették a rendszereket.

Az elkövetkező néhány évben azonban jelentős megújuláson fog átmenni az ASW, és ebben működünk együtt a Kék Óriással.

Technológiai szempontból milyenek lesznek az új rendszerek?

A. H.: Már objektumorientált eszközökkel és módon, Java nyelven állítjuk őket elő, amely megteremti a rendszerek átvihetőségének lehetőségét különböző számítógéprendszerek között. Két nagyon fontos jellemzője lesz ezeknek a fejlesztéseknek: az objektumorientáltság és az ügyfélkiszolgáló-felépítés. A Java a szakértők szerint mindkét feltételnek eleget tesz, hiszen objektumorientált nyelv, és az alkalmazások futását akár az Internet/intraneten keresztül is megengedi.

A San Francisco Project keretében az IBM és a közreműködő több szoftvercég, köztük az IBS (mint a disztribúciós, logisztikai és pénzügyi rendszerek szakértője) együtt készíti az ezeknek a követelményeknek megfelelő programkereteket. Az együttműködés célja olyan újrafelhasználható, általános, objektumorientált, előregyártott építőkövek létrehozása, amelyekből szabadon előállíthatók a legváltozatosabb igényeket is kielégítő egyedi rendszerek. Az építőkövek önmagukban és rendszerként is tetszés szerint konfigurálhatók az objektumorientált Java nyelvnek köszönhetően, amely lehetővé teszi az előre elkészített szoftveralkatrészek újrafelhasználását és a belőlük mint szabványos elemekből épülő egyedi rendszerek készítését. Ezek a programok a hordozhatóság jóvoltából — a PC-ktől kezdve az AS/400-on és Unixon át a mainframe-ekig — bármilyen számítógépen futnak majd.

A szoftverfejlesztő cégek a szabványos keretrendszereket vásárolják meg, és ezek felhasználásával építik meg saját rendszereiket, például az autóipari felhasználók számára, majd ezeket a programokat hangolják az egyes felhasználók pontos igényeihez.

Az IBS ezekben a módszerekben és az IBM szabványos fejlesztőeszközeiben látja a legbiztonságosabbnak jövőjét.

Terveink szerint 1998-tól kezdve cseréljük le meglévő és addig is folyamatosan fejlesztett rendszereinket az új keretrendszerekre alapozott objektumorientált, platformfüggetlen programcsomagra.

Mindenesetre a platformfüggetlenség érdekes jellemzője ennek a munkának egy olyan vállalatnál, amely eddig kizárólag egy cég egyetlen termékére koncentrált. Az IBS hogyan támogatja termékeinek lokalizálását?

A. H.: Elejétől fogva úgy fejlesztettük a szoftvert, hogy megvalósítottuk nemzetközi alkalmazhatóságát. Nagyon sok hasonló célú amerikai programcsomaggal ellentétben, amelyeket csak egyetlen piacra terveztek, az ASW kezeli a különféle pénznemeket, s mint már említettem, jelenleg is tíz nemzetközi változata működik.

Hogyan jellemezné az IBS értékesítési csatornájának tevékenységét?

A. H.: Megoldásszállítók vagyunk, és az eladásról is magunk gondoskodunk. A direkt értékesítési csatorna azt jelenti, hogy országonként csak egy disztribútorra koncentrálunk, amely az eladást és az üzembe helyezést egyaránt elvégzi. Központi nemzetközi osztályunk tartja a kapcsolatot az üzleti partnerekkel; ők látják el az oktatás (disztribútoron keresztül) és a hot line segítségnyújtás feladatait.

S végül, milyen új referenciákkal gyarapodtak az utóbbi időben?

A. H.: A múlt évben kötött szerződéseink közül kiemelhető az Ericsson, a Nintendo és az amerikai Detroit Diesel vállalat, amelyek mindegyike bevezette az IBS ASW-t.

KOVÁCS ATTILA

StarFire avagy Ultra Enterprise 10000

Mainframe-szintű Unix kiszolgáló

Vajon érdemes-e olyan számítógéprendszer-világújdonságot ismertetni, ami jó, ha évente egy-két példányban talál gazdára Magyarországon? Van-e különös jelentősége annak, ha egy midrange sokfelhasználós gépcsalád legnagyobb teljesítményű tagja markánsan a mainframe kategóriában jelenik meg, ráadásul úgy, hogy társaitól eltérően eredendően más szülőtől (is) származik?

A Sun Microsystems nemrég bejelentett és márciusban piacra kerülő *Ultra Enterprise 10000 Server* rendszerével kapcsolatban hangot adhatnánk további kételyeinknek is, ha a multikat és a piacot a Javával meglepő kaliforniaiak tarsolyában nem gyanítanánk ezúttal is újabb, ugyanakkor a technológiai fejlődésből logikusan következő „meglepetést". Tény, és mind több gyakorlati példa is bizonyítja, hogy a ma „adatközpontjainak" olyan kihívásoknak kell megfelelniük, amiket a vállalati szintű számítástechnika pénzügyi és architekturális szempontból növekvő igényei s az egyre rugalmasabb és menedzselhetőbb megoldások jelentenek. A Sun szerver különösen e kihívásoknak látszik megfelelni. Azoknak a Cray szervergépeknek az új generációja, amelyet a Sun a Silicon Graphicstól (SGI) vett át, amikor az SGI megvásárolta a Cray céget.

Az, hogy a Sun milyen fontosnak tartja legújabb, csúcsteljesítményű számítógépét, a bejelentést követően február elején Budapesten rendezett, előadással egybekötött bemutatón is kiderült. A világ legnagyobb szimmetrikus multiprocesszoros adatközpontjáról *Uwe Wagner*, a genfi International Enterprise Center kereskedelem-fejlesztési igazgatója és *Drew McCrocklin*, a Sun szoftverfejlesztési igazgatója tartott előadást. Mindketten a Sun Microsystems tizenöt éves fennállása egyik legkiemelkedőbb termékbejelentésének ítélték a StarFire-t. Nézzük, mire alapozták kijelentéseiket.

Igen nagy, több terabájtos alkalmazási programokhoz és olyan folyamatkritikus alkalmazásokhoz, mint az adatraktározás, nagy tömegű online tranzakció-feldolgozás, döntéstámogatás, adatelemzés, az Ultra Enterprise 10000 a mérések szerint (TPC-D) az iparág legnagyobb teljesítményét szolgáltatja. A Sun új rendszerével Unix-alapon tört be a mainframe-ek birodalmába, ahol olyan cégek szuperszámítógépeivel veszi fel a versenyt, mint az IBM, Silicon Graphics, HP és mások. A StarFire jelenleg az egyetlen Unix rendszer, amely a mainframeekhez hasonló particionálást képes biztosítani. Létrehozásával a Sun azt bizonyítja, hogy az SMP (szimmetrikus multiprocessing) rendszerek teljesítményben és ár/teljesítmény jellemzőkben felülmúlhatják az MPP (masszívan párhuzamos feldolgozású) rendszereket.

A StarFire több olyan teljesítmény-tulajdonságot tartalmaz, amely teljesen új ebben a (szerver) kategóriában. A max. 64 darab UltraSparc II processzoros támogatás — amellett, hogy több mint kétszerese a versenyképes Unix szervereknek — a Sun SMP Sparc Solaris architektúra valódi és nagymértékű skálázhatóságának az alapja. A csúcsszerver gerincét összekötő Gigaphone-XB kapcsolótábla a számítástechnikai ipar első szupergépszintű kapcsolója, amely nagyban hozzájárul a teljesítmény fokozásához, és alaposan megnövelve a belső adatkezelés sebességét egységes és konfliktusmentes elérést tesz lehetővé az egész rendszerben. A Gigaphone-XB az „adatkritikus" követelmények kielégítésének technológiai irányítása a Sun részéről. Ugyanakkor más innovatív megoldásokkal együtt azt is szolgálja, hogy az adatbázis-kezelés területén számottevően megnövekedett elvárásoknak meg lehessen felelni.

A magas fokú megbízhatóság, rendelkezésre állás, szervizelhetőség funkcióhármas, röviden a RAS, a nagy tömegű adattal, nagy sebességgel működő sokfelhasználós vállalati környezetek számára döntő. A Sun cég SunTrust robusztus RAS megoldásában majdnem folytonos rendelkezésre állás (99,99%) teljesen hardver-redundancia és masszív szoftver révén valósul meg. A RAS funkciók között található többek között a dinamikus újrakonfigurálás, „hot swap" módon cserélhető rendszerkártyák (online szerviz a működés megszakítása nélkül), kártya- és komponens-redundancia, rendszermonitor-funkciót szolgáló, mainframe szintű szolgáltatást nyújtó ún. System Service Processor és konzol, távoli online diagnosztika. Mindehhez a Sun Solaris operációs rendszerkörnyezete járul, azonnal lehetővé téve több mint 12 ezer alkalmazás portolását az Enterprise 10000-re.

Az új rendszer a tárolás szempontjából legkritikusabb területen is innovatív megoldásokat hoz: *Sun RSM Array 2000* RAID mágneslemezes tárolókat, két terabájtos mágnesszalagos könyvtárakat *(Sun Enterprise Tape Library 4/1000)*, valamint olyan adatközponttároló-üzemeltetési szoftvert *(Sun Enterprise Storage Manager)*, amely Java-alapú tárolófelügyeleti kereten alapul. A Sun nyitottságát jelzi az is, hogy skálázható tárolómegoldáscsaládját első ízben fogja támogatni heterogén platformokon (HP-UX, MS Windows NT).

K. A.

„Anyakönyvi" adatok

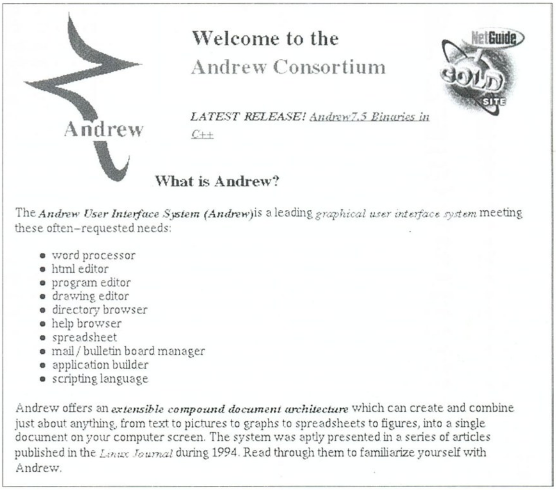
Megnevezés: Sun Ultra Enterprise 10000 Server
Fedőnév: StarFire
Architektúra: szuperskalár SPARC Version 9 UltraSparc processzorral
Processzorszám: 16–64
Gyorsítók: 16 kbájt utasítás és 16 kbájt adat a chipen; 1 Mbájt külső
CPU interfész: 64 bit Ultra Port Architecture (UPA)
Rendszeralaplapok: 4–16 db szerverenként
Operatív tár: 2–64 Gbájt, rendszerenként 512 Mbájt és 2 Gbájt tárbővítési opció
Standard interfész: SBus, 64 bit adatbusz
Tömegtároló: 30 db 2,1 Gbájt SCSI-2 diszk, illetve RSM Array 2000 max. 7x4,2 Gbájt diszk
SBus opció: Fast Ethernet-, ATM-kártya, SBus opció Ethernet vezérlő; SCSI-2; HSI; Token Ring; FDDI; Fast-SCSI-2; soros/párhuzamos optika
Szoftver: Solaris 2.5.1. op.r.; C, C++, Pascal, Fortran, Java; ONC, NFS, TCP/IP, SunNet, OSI, MHS, X.25, DCE, Novell; OpenWindows V.3 (opció); Solstice Site Manager, Solstice Domain Manager, hálózatfelügyelet a Solarishoz, Solstice DiskSuite, SparcStorage Volume Manager; SunLink IBM-kapcsolati termékek.

Az Andrew rendszer I.

Az Andrew az *Andrew User Interface System*, egy X alatti objektumorientált eszközgyűjtemény rövid elnevezése. Egységes felhasználói felületével segítséget nyújt különböző típusú szövegek, beágyazott objektumok szerkesztéséhez, levelek küldéséhez, fogadásához.

Az Andrew pályafutása 1982-ben kezdődött az egyesült államokbeli *Carnegie–Mellon* egyetemen. Ez az intézmény az IBM-mel közös fejlesztésbe fogott, melynek célja az akkori számítógépes környezet lecserélése volt egy másik, mindenki által használható rendszerre a campuson. A projekt eredményeként megszületett egy grafikus felhasználói felület, az Andrew User Interface System, és egy fájlrendszer, az *Andrew File System*. Nevüket az egyetem két fő támogatójáról, *Andrew Carnegie*-ről és *Andrew Mellon*ról kapták.

Gyakori kérdés unixos berkekben, hogy ki milyen szövegszerkesztőt használ. Nagyon sok szakmabelivel találkozni, akik a válaszként adott *ez* vagy *Andrew* szó hallatán meglepődnek. Ez a cikk éppen azért íródott, hogy tömör tájékoztatást nyújtson egy rendszerről, amit a kelleténél kevesebben ismernek.



Felépítés

Az Andrew mint *X Window System* alatt futó grafikus alkalmazás saját fejlesztésű felhasználói felülettel rendelkezik. Megjelenésében, kezelésében hasonlít a többi X alatti grafikus alkalmazáshoz. Minden Andrew ablak ugyanazokból az alkotóelemekből épül fel, így a rendszer komponensei teljesen egységes képet nyújtanak, használatuk hasonló.

Kezdetben az Andrew a fentebb említett két részből állt. Egyfelől adott volt a felhasználói felület *(VIRTUE)*, ami egy ablakkezelőt, szövegszerkesztőt és egyéb kiegészítő programokat tartalmazott, amelyek a munkaállomáson futottak. A rendszer másik oldalát a felület alatt húzódó fájlrendszer *(VICE)* alkotta; mai elnevezése az Andrew File System.

A fejlődése során kialakult feladatainak és szolgáltatásainak megfelelően a következő négy részre oszthatjuk a rendszert:

- Andrew User Environment *(AUE)*
- Andrew Toolkit *(ATK)*
- Andrew Message System *(AMS)*
- Andrew File System *(AFS)*

AUE

Az Andrew User Environment egyfajta objektumorientált programgyűjtemény. A többi között egy általános editorból (`ez`), a sokoldalú, könnyen használható help-rendszerből (`help`), egy rendszer-monitorozóból (`console`) és a shell felületből (`typescript`) áll.

ez

„EZ is Easy...”

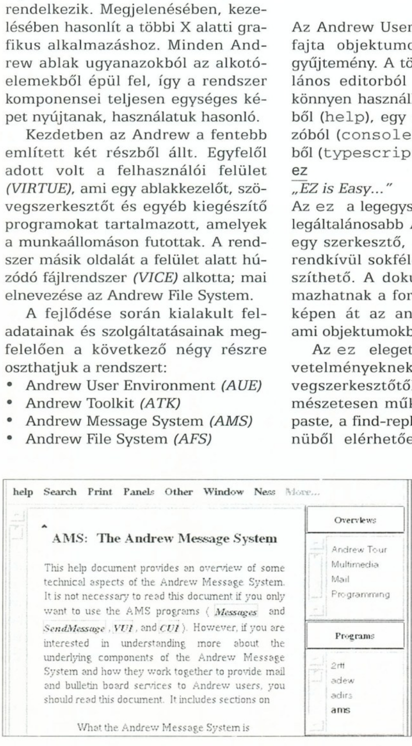
Az `ez` a legegyszerűbb és egyben a legáltalánosabb AUIS alkalmazás — egy szerkesztő, amellyel összetett, rendkívül sokféle dokumentum készíthető. A dokumentumok tartalmazhatnak a formázott szövegtől a képen át az animációig mindent, ami objektumokban leírható.

Az `ez` eleget tesz mindazon követelményeknek, amiket egy szövegszerkesztőtől elvárhatunk. Természetesen működnek a cut-copy-paste, a find-replace funkciók, s menüből elérhetően formázhatjuk a szöveget, alakíthatunk a stílusán. A bold, italic, underline stb. karakterformázás mellett változtatható a fonttípus, a szín, a méret.

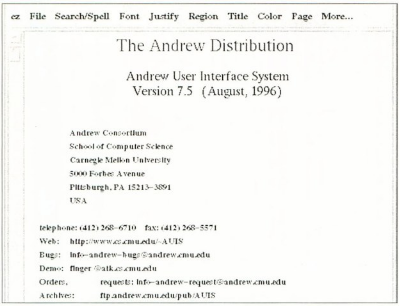
A dokumentumon lapszámozást, fejlécet és lábjegyzetet lehet elhelyezni. Ezek a szerkesztés során látszanak az adott lapon. Felhasználva a dokumentum felépítését segítő formázási lehetőségeket (chapter, section, subsection, heading, subheading, paragraph stb.) tartalomjegyzék készíthető. Ugyanilyen könnyedén, automatikusan készül az esetleges indexállomány is.

Az `ez` egy figyelemre méltó lehetősége, hogy *inset*ek, azaz beágyazott objektumok helyezhetők a szövegbe, úgymint képek, matematikai egyenletek, animációk, rajzok, táblázatok. Ezen insetek akár önálló dokumentumként is megállják a helyüket. Ebben az esetben az `ez` általános szerkesztőként, szövegeditálás helyett mint grafikus editor vagy rajzolóprogram, matematikai egyenletszerkesztő vagy éppen mint képmanipulátor használható.

Az `ez` számtalan *template*-et, stíluslapot kínál, amelyek révén a munka gyorsabbá válik. Természetesen a felhasználó maga is létrehozhat saját stíluslapot, módosított menüket stb. Ezt használja ki az `ez` forrásfájl-editorként való futtatásának lehetősége. Előre definiált stíluslapokat rendeltek bizonyos kiterjesztésekhez, amiket az Andrew felismer, és a szerkesztés során automatikusan formáz egyes kifejezéseket, szavakat stb. Így például a C forrásfájl



Az Andrew help szolgáltatása



***ez* mint szövegszerkesztő**

könnyen áttekinthetővé válik. Ugyanezen az elven alapulva a `.html` kiterjesztésű fájlokkal az `ez` html-szerkesztőként üzemel.

A kész dokumentumok ún. `ez` formátumúak. Már a szerkesztőből nyomtatható a dokumentum, de mód van más formátummá (postscript, ascii, rtf, html stb.) alakításra is az Andrew-hoz adott segédprogramoknak (például `ez2ps`, `ez2ascii`, `ez2rtf`, `ez2html`) köszönhetően.

Az `ez` szigorúan véve nem tartozik a WYSIWIG szövegszerkesztők közé. A nyomtatás, az X fontjai még beleszólnak a dokumentum végső kimenetébe, ezért — rendkívül találóan — inkább WYSLRN (What You See Looks Really Neat) szerkesztőnek nevezik.

help

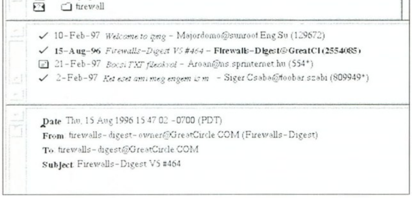
Minden program fontos része a hozzá adott kézikönyvlapok mennyisége, minősége és elérhetősége. Az Andrew e tekintetben is megfelel a kor kívánalmainak. Rendkívül részletes és szerteágazó help-rendszerében nemcsak az Andrew, hanem a Unix néhány elemének leírása is szerepel.

A `help` indításával megjelenő ablakban témakörök, programok alapján lehet válogatni, illetve egy adott kulcsszóra keresni.

ATK

Az Andrew Toolkit a cikkünk elején emlegetett felhasználói felület és programgyűjtemény, a *VIRTUE* utóda. Lényegében mára egy olyan dinamikusan betöltődő, objektumorientált és teljesen hordozható eszköztárat ad a felhasználók kezébe, ahol az objektumok akár egymásba is ágyazhatók. Ennek segítségével állíthatók elő pl. azok a multimédiás dokumentumok, ahol a különböző betűtípusok, rajzok, matematikai egyenletek, táblázatok, képek mellett akár animációk is lehetnek. Ugyanakkor ATK-val készülnek a AUE programjai, pl. az `ez` is.

Az ATK teljesen nyitott rendszer, így akár mi magunk is készíthetünk objektumokat. Ebben a munkában eszközünk az *Andrew Development Environment Workbench (ADEW)*, illetve a *NESS*. Az első egy felhasználói felületkezelő rendszer, amelylyel gyorsan készíthetők és tesztelhetők ATK alkalmazások. A NESS pedig az ATK saját programnyelve.



A *messages* ablaka

AMS

Az Andrew Message System a rendszer egyik leggyakrabban használt része. Az egyetem és az IBM közös munkájának harmadik évében készült el. A — jellemzően — ATK-val fejlesztett alkalmazás, szem előtt tartva a hordozhatóságot, a multimédia üzenetek kezelését, a többfelületes hozzáférést, képes multimédia levelek kezelésére. Az üzenetek tartalmazhatnak formázott szöveget, képeket, animációkat, táblázatokat, rajzokat stb.



Ékezetes, színes, formázott sorok a levélben

Az AMS tervezése során a nagyszámú felhasználói csoport igényeit is figyelembe vették. Ennek megfelelően a rendszer feladata többek között a BBS-elérés támogatása, magán-BBS-ek kialakítása, közös levelesládák kezelése, levelezéses szavazás lebonyolítása, levelek automatikus válogatása.

Az integrált kommunikációs rendszer kialakítása kliens-szerver alapú környezetre támaszkodik. Ennek jegyében több platformon, többfajta AMS felülettel találkozhat a felhasználó: `messages` (grafikus, teljes multimédia támogatás), `VUI` (karakterorientált felület, pl. Unixra, DOS-ra, VMS-re), `CUI` (a legegyszerűbb, sororientált felület), `Mac-Messages` (Macintosh felület).

messages

Ez nem más, mint egy multimédia kiterjesztésekkel rendelkező levelezőprogram, amely a beérkezett leveleket fogadja, válogatja, tárolja. A sima ascii és saját, Andrew mail formátuma mellett a MIME-típusú üzenetek kezelésére is alkalmas.

A `messages` mindazt tudja, amit egy levelezőprogramtól elvárhatunk. A leveleket folderekbe oszthatjuk, csoportosíthatjuk, ezekben kereshetünk, az üzenetek fájlba is menthetők vagy éppen nyomtathatók. Mód van mail-aliasok definiálására is.

Amivel több hasonló funkciót betöltő társainál, az az, hogy multimédia dokumentumok kezelésére is alkalmas, nevezetesen: formázott szöveg, kép, hang, animáció stb. továbbbítható vele.

Mind az `ez`, mind a `messages`, valamint az Andrew bemutatásra nem került részei támogatják az ISO-8859-2 karakterkészlet felhasználását, ékezetes dokumentumok készítését.

Elérhetőség, információk

1992-ben megalakult az *Andrew Konzorcium*, azzal a céllal, hogy fejlessze, bővítse és elterjessze az AUIS-t. A szervezet hivatalos home page-ének címe *http://www.cs.emu.edu/~AUIS*, ahonnan elindulva minden megtudható a programról.

Az Andrew két változatban tölthető le. A forrásszintű csomag a 6.3-as, míg a csak binárisan elérhető a 7.5-ös változatnál tart, amit már teljes egészében C++-ban fejlesztettek. A binárisan támogatott operációs rendszerek között ott a HP-UX, a SunOS és a Sun Solaris, az IBM AIX, a DEC Ultrix és természetesen a Linux.

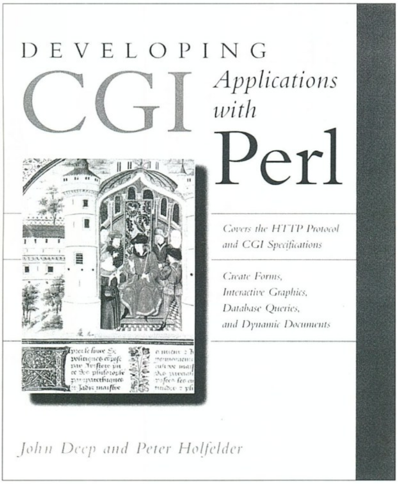
Az Andrew-t csomagokba szervezetten terjesztik, vagyis lehetőség van mondjuk csak a szövegszerkesztőt használni, a többit nem. A komplett bináris disztribúció körülbelül 10 megabájt, ami a telepítés után közel kétszer ennyi helyet foglal el a diszken.

Érdekes szolgáltatás a *Remote Andrew Demo*, amely a távoli gépen futtatva a programot, bemutatja az Andrew lehetőségeit. Ennek kipróbálásához a részletek egy `finger@atk.cs.cmu.edu` parancs kiadása után tudhatók meg.

Az alkalmazás során felmerülő kérdések, problémák megvitatására léteznek fórumok, például az *info-andrew@andrew.cmu.edu* és az *info-andrew-bugs@andrew.cmu.edu* levelezési listák (mindkettőre az *info-andrew-request@andrew.cmu.edu*-nak küldött levéllel lehet előfizetni), valamint a *comp.soft-sys.andrew@andrew.cmu.edu* newsgroup.

CZIROK LÁSZLÓ
LASZLO.CZIROK@SZTAKI.HU

„Developing CGI Applications with Perl"



Infopen az Interneten, Internet az Infopenben! Recenziós rovatunkat az éppen soros divathullámhoz igazodva, a Web és az Internet bűvöletében kezdjük újra. *John Deep* és *Peter Holfelder* könyve, amelyet a Software Station könyvespolcáról emeltünk le, a webes környezetben használható interaktív alkalmazások világába nyújt betekintést. A betekintés szó valóban találó, hiszen túlságosan szerteágazó téma ez ahhoz, hogy egy könyvben le lehessen fedni; az első alapvető bajom éppen az volt a recenzióm tárgyával, hogy mindenről megpróbál beszélni egy kicsit, ami aztán vagy elegendő az olvasónak a továbblépéshez és az érdemi munkához (azaz mindenféle színes-szagos, szív- és szerverdöglesztő webes alkalmazás összebütyköléséhez), vagy sem... — de erről majd később.

John Deep, Peter Holfelder: „Developing CGI Applications with Perl", Wiley Computer Publishing, 1996, ISBN 0-471-14158-5

Noha a profi „neccmesterek" (pardon, webmasterek) már relációs adatbázis-kezelőkkel integrált Web alkalmazásokkal dolgoznak, a túlnyomó többség számára, amelyiknek legfeljebb a saját vagy kisvállalkozása „hómpéjdzsének" az összeeszkábálása a cél, még belátható ideig a könyvben ismertetett, HTML–CGI–Perl trió biztosítja a legrövidebb utat a cél eléréséhez. A HTML rövidítés magyarázatával nem is óhajtjuk megsérteni olvasóinkat, kis hazánkban ezt nem ismerni már egyenesen olyan műveletlenségnek tűnhet fel, mintha valaki nem fújja kívülről az aranycsapat névsorát, vagy nem hallott a Tocsik-ügyről...

Nem így áll a helyzet a CGI-vel, amellyel a neccet csupán léha szörfözésre használók legfeljebb akkor találkoznak, ha lányos zavarukban a „Show Source" gombra tenyereltek, s mindenféle ismeretlen parancsok sejlenek fel képernyőjükön. A CGI a Web szerverek többé-kevésbé szabványos interfésze, mely — többek között – interaktív HTML dokumentumok létrehozását teszi lehetővé, a szerveren futtatandó programok segítségével: ha valamilyen Web címen kérdőívet töltünk ki, vagy adatokat írunk be, nagy valószínűséggel CGI scripttel van dolgunk a vonal túlsó végén.

Ami a trió harmadik tagját illeti, nos, a Perlt nem ismerni nem bűn, hanem (sajnálatos) állapot — e programnyelv ugyanis egészen a legutóbbi időkig csak a unixos rendszergazdák körében volt ismert és kedvelt, egyszerűsége és gazdag szolgáltatásai révén. Éppen ezen tulajdonságai teszik ideálissá a CGI interfész által szolgáltatott információk szerveroldali feldolgozására, a CGI scriptek implementálására, s így vált a Web fejlesztők egyfajta közös nyelvévé (legalábbis a szekértábor unixos oldalán).

A szerzők dicséretes módon próbálták mindezt összefogni a könyv lapjain, felemás eredménnyel. A legtöbbet az alapvető HTML parancsokról tudhatunk meg, csak hát ez az az információ, amit amúgy a legkönynyebben megszerezhet az ember. A CGI és a HTTP protokoll ismertetése ezzel szemben már jóval elnagyoltabbra sikeredett, a Perl bemutatása helyett pedig csak — amúgy igen hasznos — linkeket és referenciákat kapunk. (A továbbiakban — talán kárpótlásképpen — az egyes alkalmazások ismertetésénél az azokhoz szükséges Perl alapismereteket is megadják a szerzők, mely megoldás ugyan didaktikailag szerfelett helyteleníthető, viszont leplezetlen módon sarkallja a köny végigolvasására az olvasót...)

A későbbi fejezetekben egy-egy konkrét alkalmazás kapcsán nyerünk bevezetést a webes applikációk rejtelmeibe, s ezek a részek jóváteszik a bevezetés hiányosságait. Az ismertetett, időnként meglepően komplex alkalmazások (pl. interaktív formanyomtatványok, számlálók, projektmenedzsment rendszerek, adatbázis-kezelő interfészek stb.) olyan „archetípusok", amelyekkel mindenkinek dolga lesz, ha webezésre adja a fejét.

Összességében azt mondhatom, mire a könyv végére értem, már többé-kevésbé megbékéltem a fentiekben megmorgott hiányosságaival: ha minden benne lenne, amit hiányoltam, inkább az Encyclopaedia Britannicára hajazna, legalábbis terjedelemben. Így viszont bele kell nyugodni abba, hogy önmagában nem elég a Web meghódításához, s kezdőknek mindenképpen javasolnám valami bevezető jellegű Perl és HTML könyv elolvasását is — magyarán szólva, referencia helyett inkább egyfajta „receptkönyvként" használandó; annak viszont jó szívvel ajánlható.

Amit viszont nagyon rossz néven vettem, az nem a szerzők hibája, inkább a kiadóé: a hátsó borítólapon nagy betűkkel reklámozzák, hogy a kiadó ilyen és ilyen URL címén (nem titok, *http://www.wiley.com/compbooks*) hozzáférhetőek a könyvben ismertetett CGI mintaprogramok. Nosza rajta, lássuk — fél óra keresgélés és a legeldugottabb hivatkozásokra való bekukkantás után feladtam, s írtam egy enyhén kritikus levelet a kiadó E-mail címére; erre ugyan jött válasz, de nem volt benne sok köszönet, csak arról értesített egy nyájas, Mailer Daemon nevű illető, hogy levelem az előfizetés hiánya miatt kézbesíthetetlen. Hogy ki és milyen előfizetéssel hibádzik, azóta se tudom, s bevallom, már nem is nagyon érdekel, az élet szép, a gépelés egészséges és edzi az ujjakat...

BARTÓK NAGY JÁNOS

Gépkarton



A hogy elnézem, előbb-utóbb kezdenek bejönni az öreg Bangemann megérzései. Még 1993-94 táján, az információs társadalom jövőjét taglaló munkájában, amelyet egyszerűen csak Bangemann-jelentésnek szoktak emlegetni (Recommendations to the European Council Europe and the global information society), vetette fel annak lehetőségét, hogy akár nagyobb méretekben is távdolgozhassunk, mármint a nagy hálózaton keresztül. Azt is megpróbálta előre jelezni, hogy 1996 végére a fehérgalléros munkások két százaléka fog teledolgozni, és ez hatalmas befektetési lehetőségeket kínál a magánszektornak is.

Kíváncsi lennék rá, hogy ez a két százalék mennyire jött be az Európai Közösségben és Magyarországon. Mintha egy kicsit még elmaradnánk ettől az óriási fellendüléstől, vagy lehet, hogy nem is maradtunk le annyira, hanem inkább csak szeretjük a saját módszereink szerint megvalósítani a dolgokat. (A boltokban/vendéglőkben/Internet- és távközlési szolgáltatóknál az árak már a nyugati piacgazdasághoz közelítenek, de a user interfész inkább hagyományos. Még ha írásban és szóban is megrendel az ember bármit, esetleg többször meg is sürgeti, akkor sem izgulnak be a nagy biznisz reményében.)

Bangemann azt is megjövendölte, hogy bármennyire el fog terjedni a hálózatok használata, a munkaképes lakosság közel egyharmada nem lesz képes használni az új eszközöket. Azt hiszem, ebben jobban közelítünk a világszínvonalhoz.

De váltsunk át pozitív példákra. Rulez-díj győztesünk, *Ballner György* karikaturista *(http://www.sprinternet.hub/~ballner)* mintha a Bangemann-hívők egyike lenne, hiszen meglehetős alapossággal kiaknázta a kibertér számára rendelkezésre álló lehetőségeit, szem előtt tartva a honi körülményeket, úgymint sávszélesség, fizetési módok, garanciák.

Először is a távmunka/telemunka egyik kézenfekvő lehetősége: a művész megkapja a megrendeléseket a hálózaton keresztül, itt részletesen specifikálja, hogy amennyiben fotó alapján kérünk karikatúrát, mekkora méretű képet küldjünk. Ezután meg is kaphatjuk az áhított művet a kívánt minőségben, amint megérkezett az átutalásunk a megadott helyre. Itt megfigyelhetjük, milyen szofisztikált is ez a módszer: ismerve a honi viszonyokat, nem kér hitelkártyaszámot, bankgaranciát, viszont szintén a helyi tapasztalatok birtokában abból-baj-nem-lehet alapon csak akkor veszi komolyan a megrendelést, ha az átutalás befutott.

További lehetőség gyűjtőknek és rajongóknak, hogy akár a mester kezével rajzolt példányhoz is hozzájuthatunk, ami jogos, hiszen a sokszorosító technikákkal készült alkotások mindig olcsóbbak szoktak lenni az egyedieknél. Nem árul zsákbamacskát: már a nagy megrendelésünk előtt kellő refenciákat kaphatunk, hiszen az eddig elkészült alkotások közül megtekinthetünk néhányat. A munka otthon végezhető, saját időbeosztással. A rendszer nagyon jól továbbfejleszthető, például a virtuá-

Rulez-díj

1996. szeptemberi számunktól együttműködünk az iNteRNeTTo-val (www.idg.hu/internetto/cyber/halorago/) a díjazottak bemutatásában. Havonta öt kiadványt kap szavazásra a Rulez-díj zsűrije: Bertók Attila, Gerényi Gábor, Jyrki Halonen, Pogány György és Szabó Tibor. Az elbírálás során azokat a szempontokat mérlegelik, hogy a kiadványok mennyire használják ki az új média lehetőségeit, milyen a design, a tartalom, és hogyan integrálódik a világhálózatba. A Rulez-díj szponzora, az Inter-Európa Bank a klasszikusnak nevezhető home-banking segítségével valósítja meg azt, hogy ügyfelei bármikor, bárhonnan intézhessék banki ügyleteiket. A Rulez-díj jutalma egy Inter-Európa telefonbank bon 20 000 Ft értékben, amely bármelyik Inter-Európa Bank-fiókban átvehető. A rulez legszabatosabb magyar fordítása: király. Régen úgy mondtuk volna: fenséges, kevésbé régen: szuper. Ellentéte a suxx, a szívás, a vacak, az ócska, a gagyi, a zizi. Rulez kifejezésünk az angol rulesra vezethető vissza, ami annyit tesz: uralkodik. Unix rules the world, mondták a guruk, vagyis a Unix operációs rendszer a legjobb a világon. Ebből torzították az amerikai hálórágók a rulezt.

Januári jelöltek voltak

Andrassew Iván (kísérleti) hírlevele *(www.elender.hu/˜andrasse/)* Megrendelhetjük az író folytatásos életműkiadását: csak egy levelet kell írnunk: andrasse@mail.elender.hu, és ő hetente-kéthetente postázza írásainak gyűjteményét. Az egykori egyszemélyes folyóiratok szimpatikus utóda. A cikkek egy része a Hócipőben, más része a Színes Vasárnapban jelenik meg, vagy a Rádióban hangzik el, így a hírlevél nevettet és elgondolkodtat, a megfelelő arányokban.

Ballner *(www.sprintermet.hu/˜ballner/)* A népszerű karikaturista nemcsak rajzait tette közzé az Interneten, de egy különleges szolgáltatást is nyújt. Ha elküldjük neki fényképünket, néhány nap múlva meglep minket a karikatúránkkal. (Kár, hogy a kezdőlap lassan jön be a táblázat miatt.)

CrashCourse *(http://www.idg.hu/internetto/light/course/index.html)* Török András budapesti útikönyvének angol nyelvű változata, Eperjesi Ágnes fotóival. A szerző a fórumban válaszolja meg a külföldi érdeklődők kérdéseit is.

A németországi szállodaportások „Aranykulcs" egyesülete *(members.aol.com/Conc4Hilt/conchun.htm)* szászországi szekció. Akinek ehhez különösebb kommentár kell, az még nem látott szállodaportást. A lap amúgy tisztességesen van összeállítva: a hivatalos infók mellett Drezda városának nevezetességeit is megismerhetjük, sőt, foglalhatunk jegyet az Operába is.

Software Station *(sws.xco.hu)* Internet-könyvesbolt. 23 000 számítástechnikai könyvből válogathatunk. Bibliográfiaként is használható lista, persze az ismertetőkkel együtt lenne igazi. A kiválasztott könyveket meg is rendelhetjük. Megnézhetjük az új könyvek listáját, és kereshetünk az adatbázisban cím vagy szerző szerint, ínyencek ISBN szám szerint. A listát rendszeresen frissítik.

NYÍRŐ ANDRÁS



lis budapesti városnézések honlapján a Vörösmarty térre, ahol a portrérajzolók szoktak ücsörögni — csak egy linket kell elhelyezni, és máris éjjel-nappal, fagyban, hóban, szélben kérhetünk portrét magunkról, anélkül, hogy kimozdulnánk. A művészeknek sem kell tülekedniük, hogy a megfelelő helyre tehessék székeiket és rajztábláikat, megférnek akárhányan a hálózaton. Nincs többé harc az erős fiúkkal, akik mindig felajánlják, hogy megvédik az ember boltját, vendéglőjét, feltehetően van még egypár évünk addig, amíg a hálózaton is utána tudnak majd menni az utcai művészeknek.

Ha valaki mégis vonzódik a műalkotások instant elkészítéséhez, akkor csak utcai terminálokat kell üzembe helyezni a látogatottabb helyeken, kamerával felszerelve. Az utánunk következő generációk már meglepően jól tudják követni a világ fejlődését: a multinacionális gyorséttermi hálózatok üzleteiben néhány helyen felfedezhetünk olyan számítógépeket, amelyeken nincs billentyűzet, hanem csak érintésérzékeny képernyő. Megnézhetjük rajta a moziműsort, más kulturális programok közül is válogathatunk, és persze ott vannak a szponzorok hirdetései is.

Eddig nincs is ebben semmi különös, néhány hete azonban a következőt tapasztaltam: egy kissrác, lehetett vagy százhúsz centi magas, nem érte el a kihelyezett terminál képernyőjét, lentről navigált a haverjának, aki ugyan már elég magas volt, de láthatóan nem sokat értett az egészből, csak huzigálta az ujját a képernyőn, ahogy a földszintes diktálta. Mégiscsak beindul itt majd az élet, ha ezek felnőnek...

SZABÓ TIBOR

Pályázat

Az iNteRNeTTo pályázatot hirdet a következő kategóriákban:

- novella (max. 5000 karakter);
- riport (max. 12 000 karakter);
- közgazdasági elemzés (max. 30 000 karakter).

Olyan műveket várunk, amelyek módszerükben vagy témájukban kapcsolódnak az Internethez. Szakmai zsűri bírálja el a pályamunkákat. A fődíj mindhárom kategóriában 25 000 Ft, a 2–3. díj 10 000 Ft. A megfelelő színvonalú írásokat az iNteRNeTTo közli. Az újságírói kategóriában közönségszavazati díjat is kiadunk (15 000 Ft). Határidő: április 30., eredményhirdetés: május 15. E-mailben a *nyiro@internetto.hu* címre, vagy postán a szerkesztőség címére lehet beküldeni a pályaműveket: 1012 Budapest, Márvány utca 17. További részletek az Interneten: *http://www.internetto.hu.*

Nyílt rendszeres eseménynaptár

Az Infopen Online (www.eunet.hu/infopen) Naptár rovata további adatokkal és hiperlinkekkel együtt, folyamatosan frissítve sorolja fel a közeljövőben megrendezésre kerülő nyílt rendszeres eseményeket

Március

10–14.: UniForum '97 és MMTA's UNICOM '97
San Francisco, Moscone Center
13–19.: CeBIT '97 Hannover
24–25.: DBFORUM Budapest, rendező: MMI
27. HTE szakosztályülés
A MAGIC és az Informix objektumorientált adatbázis-kezelő architektúrái Budapest, PKI Konferenciaterem
Rendező: HTE Számítástechnikai Szakosztály

Április

7–12.: Sixth International World Wide Web Conference
10–12.: VIII. Országos Reklámkonferencia Kecskemét
14–19.: Hannoveri Ipari Vásár
17–18.: Open to the WWWorld Zágráb
22–25.: COMDEX/UK '97 London

Május

6–10.: Ifabo Budapest '97
12–13.: NETFORUM+ATMFORUM Budapest, rendező: MMI
27–31.: Industria Budapest, BNV
27–29.: Networkshop '97 Keszthely
28–29.: EDI '97

Június

2–4.: DECUS Magyarország Konferencia Eger
2–5.: COMDEX Spring '97 és Windows World Atlanta, Georgia World Congress Center
4–7.: KRIMINALEXPO '97
Budapest Kongresszusi Központ
10–13.: KOMMUNÁLEXPO Budapest, BNV
17–20.: Object World/United Kingdom
London

Július

6–8.: AusWeb '97 Ausztrália

Augusztus

20–23.: EUROLOGO '97 Budapest

Szeptember

8–12.: EUROGRAPHICS Conference és CAMP Trade Show Budapest
12–21.: Budapesti Nemzetközi Vásár

Szerkeszti: Hutter Ottó
(269-8272, otto.hutter@sztaki.hu)

infopen®

Nyílt rendszerek magyarországi hírmagazinja
Kiadja az **OpenInfo Kiadó**
Felelős kiadó: **Dr. Vas Zoltán**
Alapító főszerkesztő: **Kovács Attila**
Szerkesztőbizottság:
Dr. Demetrovics János, Nagy Miklós, Dr. Remzső Tibor, Dr. Sima Dezső, Dr. Telbisz Ferenc
Főszerkesztő: **Dr. Hutter Ottó**
Olvasószerkesztő: **Gams Judit**
Művészeti és műszaki vezető (fotó): **Szabó Tibor**
Titkárságvezető: **Polyák Erzsébet**
Nyomás és kötés: **Akadémiai Nyomda**
Felelős vezető: **Freier László**
Levilágítás: **LaserGraph**
A cikkekben és táblázatokban szereplő adatokat gondosan ellenőrizzük. Az esetleg mégis előforduló pontatlanságokért és tévedésekért azonban a kiadó nem vállal felelősséget.
Előfizetés:
az **OpenInfo** kiadónál
egy évre: 1900 Ft + áfa
Telefon: 166-5644/447, 413;
fax: 166-7503;
postacím: 1111 Budapest, Kende u. 13.
Internet címek: infopen@ind.eunet.hu, http://www.eunet.hu/infopen
Hirdetésfelvétel:
Pap Katalin, Árvai Katalin
Tel.: 214-9492, 156-3211/168, 200 Fax: 214-9492, 156-3211/201
E-mail: alaplap@mail.datanet.hu

HU ISSN 1217-1905



A Sun Ultra Enterprise szervereknek köszönhetően

új időszámítás kezdődött a nagyvállalati hálózati számítástechnikában!



Mindig gondot okoz, ha a cég növekedése a meglévő szerver lecserélését teszi szükségessé. Erre a problémára a legbiztosabb megoldás a Sun Ultra Enterprise szervercsalád, mely a mainframe-képességeket a nyílt rendszerek előnyeivel ötvözi. Ezeket a szervereket úgy tervezték, hogy a vállalat igényeinek növekedésével párhuzamosan a kapacitásuk is növelhető legyen, akár több terabyte nagyságú adatbázisokról, akár többezer felhasználó egyidejű támogatásáról van szó. A moduláris, skálázható felépítés lehetővé teszi a CPU és az I/O sávszélesség igény szerinti bővítését 30 processzorig, 30 GB memóriáig, mainframe-szintű I/O képességekig, továbbá 10 TB tárolási kapacitásig. A rugalmas növekedés további garanciáját jelenti a Sun Solaris operációs rendszere. A vállalat létfontosságú alkalmazásainak megbízható, leállás nélküli működését szolgálja a redundáns tervezés, a kikapcsolás nélkül cserélhető alkotóelemek, a SunService SunSpectrum csomagja – amely rendszermegközelítésű, teljes körű hardver és szoftver támogatási szolgáltatást és verziókövetést biztosít –, valamint a világ egyik legtapasztaltabb és legnagyobb UNIX-szakembergárdája. Elérkezett tehát a vállalati hálózati számítástechnika korszaka, amikor a Sun Ultra Enterprise szerverek segítségével a nagyvállalatok számítástechnikai igényei – a növekedés ütemének megfelelően – értékállóan elégíthetők ki. További információért forduljon a hálózati számítástechnika szakértőjéhez, a Sun-hoz!

Sun
microsystems

THE NETWORK IS THE COMPUTER™

Sun Magyarország Kft. 1027 Budapest, Kapás u. 11-15. Tel.: 202-4415, Fax: 201-2731, WWW-cím: http://www.sun.hu